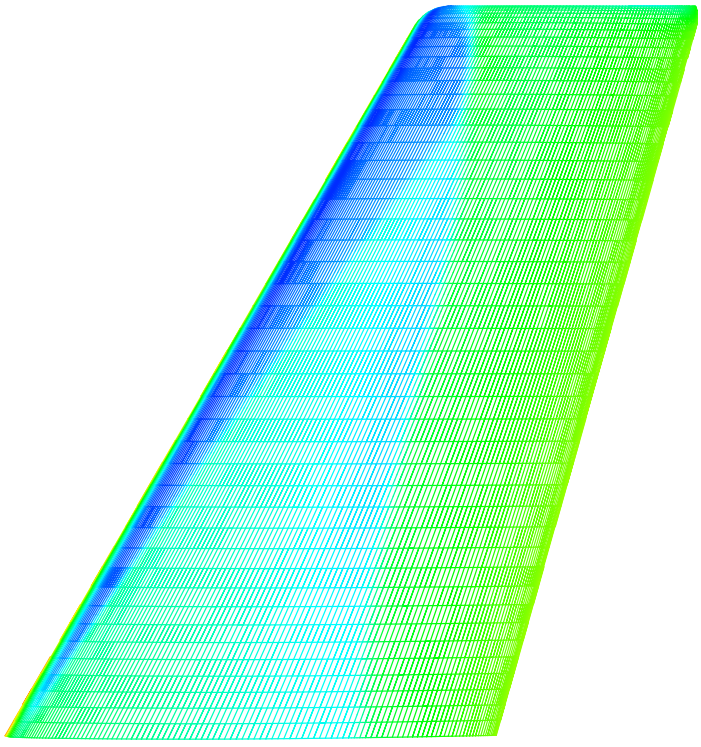
FOI-R--0298--SE
December 2001
ISSN 1650-1942

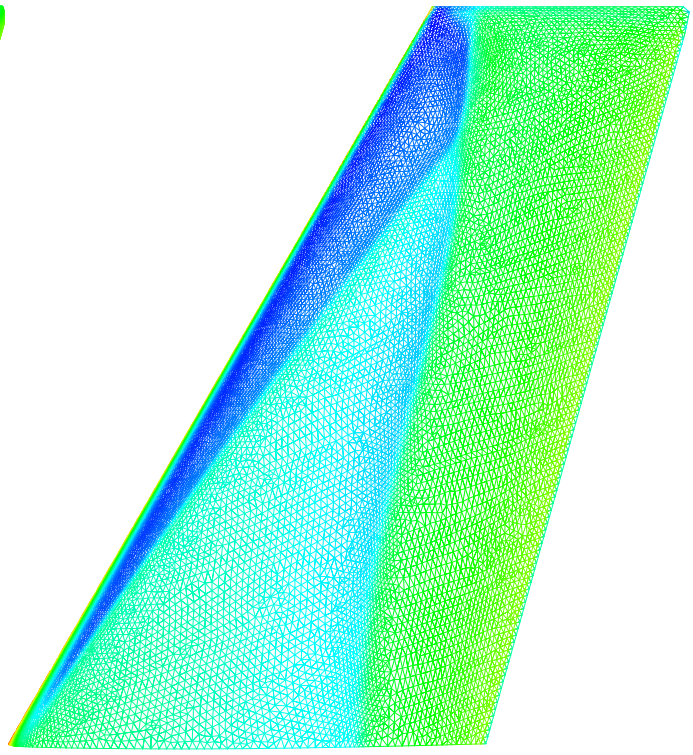
Scientific report

Peter Eliasson

EDGE
A Navier-Stokes Solver for Unstructured Grids



EURANUS



EDGE

Division of Aeronautics, FFA
SE-172 90 STOCKHOLM

FOI-Swedish Defence Research Agency
Division of Aeronautics, FFA
SE-172 90 STOCKHOLM

FOI-R--0298--SE
December 2001
ISSN 1650-1942

Scientific report

Peter Eliasson

EDGE
A Navier-Stokes Solver for Unstructured Grids

Division of Aeronautics, FFA
SE-172 90 STOCKHOLM

Abstract

This report describes the compressible Navier-Stokes solver *EDGE* for unstructured grids. The solver is based on an edge-based formulation for arbitrary elements and uses a node-centered finite-volume technique to solve the governing equations. Two spatial discretizations of the convection terms are described, compact discretizations of the thin-layer and fully viscous terms have been proposed and evaluated. The governing equations are integrated explicitly towards steady state with Runge-Kutta time integration. The convergence is accelerated with agglomeration multigrid and implicit residual smoothing.

A validation is carried out in two and three dimensions for external flows. The validations focus on comparisons between *EDGE* and the cell centered solver *EURANUS* on structured grids. Also the effect of different types of elements are investigated.

The results with the unstructured and structured approach compare well for all cases. The rate of convergence is comparable although higher CFL numbers can be used with the structured solver. The robustness of the unstructured solver is at least as good as with the structured solver.

Two main differences are found. The first is that the decay of the maximum total pressure loss for subsonic Euler calculations is approximately second order accurate as the grid is refined for the node-centered scheme but only first order using the structured cell centered approach. The second difference concerns the flow over an airfoil at a low Reynolds number and no artificial dissipation. Here the rate of convergence is much slower with the unstructured approach. Small oscillations in the pressure can also be observed in the nose region.

Table of Contents

1 Introduction

The need and demand for accurate and rapid tools for solving CFD problems in industrial applications is high. The approach for more and more complex geometry in CFD problems has forced the grid-generation techniques towards nearly automized algorithms producing unstructured grids, hence, methods for flow computations on such grids are required.

To meet these requirements, FFA has in a first stage developed an Euler solver for unstructured grids called *EDGE* [1]. This report concentrates on the extension to viscous and turbulent flow. Although the Euler part has been described and validated earlier [1], the convective part of the solver is described in this report for completeness. Also, the modifications due to stretched grids has been introduced.

EDGE uses a node-centered finite-volume technique to solve the compressible equations. It relies on explicit time integration with convergence acceleration by agglomeration multigrid and implicit residual smoothing. The considered spatial discretizations are all formally second order accurate in space. Two discretizations on the convections are described, one central scheme with artificial dissipation and one upwind scheme. Two compact discretizations of the viscous terms are described and evaluated. One discretization uses a compact discretization of the normal derivatives leading to a thin-layer approximation. The additional tangential derivatives may be computed from node-based gradients and added for a fully viscous contribution. The second discretization makes use of corrected gradients computed with a Green-Gauss approach. The difference between the methods is shown in the computations.

The *EDGE* code is denoted “grid-transparent” since it is equally applicable to any type of element like hexahedral, triangular prisms and tetrahedral elements in 3D and quadrilateral and triangles in 2D. An edge-based formulation of the solver and a preprocessor[2] that translates element-based information to edge-based information allows the system to handle structured, unstructured and hybrid grids seamlessly. Only node and edge data are used in the solver.

The validation is carried out in two dimensions for external flows over airfoils and for a wing in three dimensions. An Euler case, a laminar low Reynolds number case and a two turbulent cases have been used. More Euler validations have earlier been carried out [1]. The validations focus on comparisons between *EDGE* and the cell centered solver *EURANUS* [3] on structured grids, often on the same grid in an unstructured format but also on a completely unstructured grid. In some cases the structured grid in two dimensions has been triangulated to see the effect of different cell types.

2 The Compressible Navier-Stokes Equations

2.1 Laminar Navier-Stokes Equations

The Navier-Stokes equations which govern viscous compressible flow in three dimensions can be written in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega}\bar{q}\,dV + \oint_{\partial\Omega}\dot{f}(\bar{q},\bar{n})\,dS = \oint_{\partial\Omega}\dot{f}_V(\bar{q},\bar{n})dS \tag{2-1}$$

with

$$\bar{q} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{pmatrix}, \dot{f}(\bar{q},\bar{n}) = F\cdot\bar{n} = \begin{pmatrix} \rho u_n \\ \rho u_n u_1 + p n_x \\ \rho u_n u_2 + p n_y \\ \rho u_n u_3 + p n_z \\ u_n(E+p) \end{pmatrix} \tag{2-2}$$

where $u_n = \bar{u}\cdot\bar{n}$, Ω is an arbitrary finite region with boundary $\partial\Omega$, $\bar{q}$ the conservative variables, $F\cdot\bar{n} = \dot{f}(\bar{q},\bar{n})$ the convective flux.

The viscous flux is

$$\dot{f}_V(\bar{q},\bar{n}) = F_v\cdot\bar{n} = \bar{F}_{v1}n_x + \bar{F}_{v2}n_y + \bar{F}_{v3}n_z \tag{2-3}$$

where

$$\bar{F}_{vi} = \begin{pmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ u_j\tau_{ij} - q_i \end{pmatrix}. \tag{2-4}$$

The stress tensor τ can be written, using tensor notation, as

$$\tau_{ij} = \mu\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}(\nabla\cdot u)\delta_{ij}\right] \tag{2-5}$$

where μ is the dynamic viscosity and the heat flux q_i is written as

$$q_i = -\kappa\frac{\partial T}{\partial x_i}. \tag{2-6}$$

The thermal conductivity is found from the viscosity μ and the constant Prandtl number Pr

$$\kappa = \frac{\mu C_p}{Pr} \tag{2-7}$$

where C_p is the specific heat coefficient under constant pressure.

For a calorically perfect gas the following relation holds for the total energy E:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho|\bar{u}|^2 \tag{2-8}$$

Equations (2-1) - (2-8) form a closed set of 5 equations with 5 unknowns. The variables actually stored in the computer code are the primitive variables $\bar{v} = (\rho, u_1, u_2, u_3, p)^T$.

2.2 Reynolds Averaged Equations

Using a k-ω turbulence model the Reynolds averaged Navier-Stokes equations can be written as

$$\frac{\partial}{\partial t}\int_{\Omega} \bar{q}\, dV + \oint_{\partial\Omega} \bar{f}(\bar{q}, \bar{n})\, dS = \oint_{\partial\Omega} \bar{f}_V(\bar{q}, \bar{n}) dS + \int_{\Omega} S_t\, dV \tag{2-9}$$

with

$$\bar{q} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \\ \rho k \\ \rho \omega \end{pmatrix}, \bar{f}(\bar{q}, \bar{n}) = \begin{pmatrix} \rho u_n \\ \rho u_n u_1 + p^* n_x \\ \rho u_n u_2 + p^* n_y \\ \rho u_n u_3 + p^* n_z \\ u_n(E + p^*) \\ \rho u_n k \\ \rho u_n \omega \end{pmatrix}, \tag{2-10}$$

$$\bar{F}_{vi} = \begin{pmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ u_j\tau_{ij} - q_i + (\mu + \sigma_k\mu_T)\dfrac{\partial k}{\partial x_i} \\ (\mu + \sigma_k\mu_T)\dfrac{\partial k}{\partial x_i} \\ (\mu + \sigma_\omega\mu_T)\dfrac{\partial \omega}{\partial x_i} \end{pmatrix}, \bar{S}_t = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ S_k \\ S_\omega \end{pmatrix}. \tag{2-11}$$

The model constants σ_k, σ_ω and the source terms depend on the variant of the k-ω turbulence model. In this case the standard model by Wilcox[4] has been used where $\sigma_k = \sigma_\omega = 0.5$ and in which the source term is given as

$$\begin{bmatrix} S_k \\ S_\omega \end{bmatrix} = \begin{bmatrix} P - \beta_s\rho k\omega \\ \alpha P\omega/k - \beta\rho\omega^2 \end{bmatrix} \tag{2-12}$$

where $\beta_s = 0.09$, $\alpha = 0.556$, $\beta = 0.075$ and where the production P is given by

$$P = -\left(\frac{2}{3}\rho k\delta_{ij} - \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}(\nabla \cdot u)\delta_{ij}\right)\right)\frac{\partial u_i}{\partial x_j}. \tag{2-13}$$

p^* is the static pressure with added turbulence

$$p^* = p + \frac{2}{3}\rho k. \tag{2-14}$$

The stress tensor τ can be written as

$$\tau_{ij} = (\mu + \mu_T)\left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}(\nabla \cdot u)\delta_{ij}\right] \tag{2-15}$$

and the heat flux q_i is written as

$$q_i = -(\kappa + \kappa_T)\frac{\partial T}{\partial x_i}. \tag{2-16}$$

The turbulent viscosity μ_T and the turbulent thermal conductivity κ_T are defined as

$$\mu_T = \frac{\rho k}{\omega} \tag{2-17}$$

and

$$\kappa_T = \frac{\mu_T C_p}{Pr_T} \tag{2-18}$$

respectively where $Pr_T = 1$ is the constant turbulent Prandtl number. For a calorically ideal gas the following relation holds for the total energy E:

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho \bar{u}^2 + \rho k \tag{2-19}$$

In this report in particular results with the k-ω turbulence model are presented but a few results with an explicit algebraic Reynolds stress model (EARSM) based on the k-ω model are presented as well. For further details on the EARSM mode see Wallin *et al.* [5].

3 Geometrical Considerations

The finite-volume technique requires control volumes surrounding the unknowns in the nodes. The control volumes are non-overlapping and are formed by the dual grid obtained from the control surfaces at the edges. The dual grid is supplied by the preprocessor [2] as an input to the flow solver. The grid with its dual grid is depicted in two dimension in Figure 3-1.

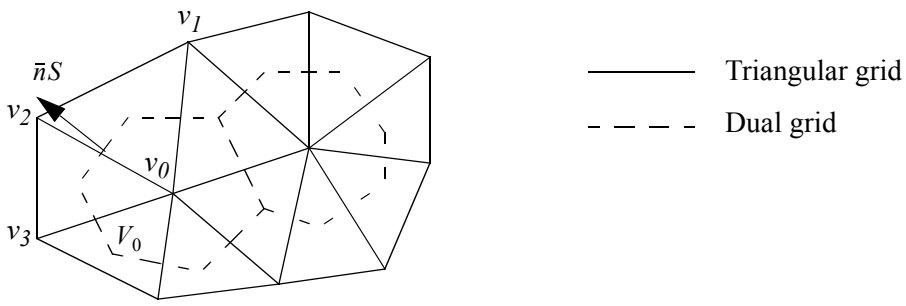


Fig. 3-1 : The input grid (triangular grid) and its dual grid forming the control volumes.

The coordinates of the input grid are provided for each node and the connectivity is supplied in an edge based manner where an edge connects two nodes. In addition to the node numbers a control surface $\bar{n}S$ is supplied for each edge. The control surfaces of all edges emerging from a node enclose the control volume of a node. The surface is given as a vector with direction $\bar{n}$ with its size S as magnitude. In 2D a length is supplied and the vector has only two components. If the edge is denoted $e = (v_i, v_j)$ then the surface associated with the edge forms a sharp angle $< 90°$ to the edge vector $\bar{x}(v_j) - \bar{x}(v_i)$, also denoted $\bar{x}_j - \bar{x}_i$.

If each control surface is considered to lie in a plane, the volume of the cone formed by the control surface and one of the nodes is

$$\frac{1}{2\,ndim}\bar{n}_{0k}S_{0k}\cdot(\bar{x}_k-\bar{x}_0) \tag{3-1}$$

where $ndim$ is the space dimension. Quantities with two sub indices denote edge quantities where the indices denote the two nodes connecting the edge.

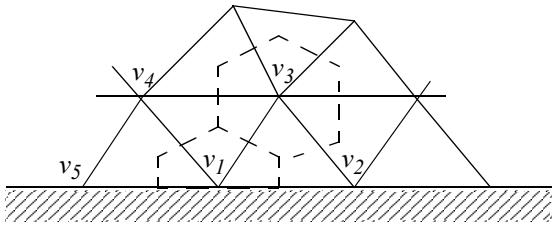
All edge volumes of a node sum up to the control volume, i.e.

$$V_0 = \frac{1}{2\,ndim}\sum_{k=1}^{n_0}\bar{n}_{0k}S_{0k}\cdot(\bar{x}_k-\bar{x}_0)\ . \tag{3-2}$$

n_0 is the number of neighbor nodes to node v_0 or, equivalently, the number of edges connected to node v_0.

To close the control volumes at boundaries, control surfaces are supplied at the boundary. In Figure 3-2 the control surfaces to node v_3 are given at all edges connected to v_3, whereas the edges to the boundary node v_1 do not close the control volume. To close it, a control surface is given separately for the boundary node v_1.

Fig. 3-2 : Control volumes at an inner node and a boundary node.



At a corner point where two or more boundaries meet the boundary control surface is split into control surfaces for each boundary condition separately. In that case, the boundary node may occur in several boundary conditions.

In addition to the control surface supplied for each boundary node, an inner point is also supplied at all boundaries. The inner nodes is used in some of the boundary conditions described in Section 6 on page 29 and the inner node is chosen as the end node of the contiguous edge most orthogonal to the boundary surface. In the example above, node v_3 is the inner node to node v_1.

4 Spatial Discretization

The finite volume discretization is obtained by applying the integral formulation of the governing equations in Equation (2-9) to the control volume surrounding the unknown $\bar{q}_0$ at node v_0:

$$\frac{\partial}{\partial t}(\bar{q}_0 V_0) + \sum_{k=1}^{n_0} \dot{f}_{0k} S_{0k} = \sum_{k=1}^{n_0} \dot{f}_{V0k} S_{0k} + S_{t\,0} V_0 \tag{4-1}$$

where n_0 is the number of neighbors to node v_0. The surfaces S_{0k} enclose the control volume for node v_0 and form the dual grid illustrated in Figure 3-1 in 2D for a given triangulation. The flux vectors $\dot{f}_{0k}$ and $\dot{f}_{V0k}$ are computed on the edge consisting of nodes v_0 and v_k where S_{0k} is given, the source term $S_{t\,0}$ is computed directly at the node.

4.1 Convective Terms

The schemes for the convective flux $\dot{f}_{0k}$ considered here are based on a central discretization with dissipation terms of either artificial dissipation type or upwind flux difference splitting type.

4.1.1 Central scheme with artificial dissipation

The convective flux across the cell face between nodes v_0 and v_1 is computed as

$$\dot{f}_{01} = \dot{f}\left(\frac{\bar{q}_0 + \bar{q}_1}{2}\right) - \bar{d}_{01} \tag{4-2}$$

where $\bar{d}_{01}$ denotes the artificial dissipation. A blend of second and fourth differences are chosen as artificial dissipation, this corresponds to a blend of first and third differences for the fluxes. The following form has been shown to be suitable[6],[7]:

$$\bar{d}_{01} = (\varepsilon_{01}^{(2)}(\bar{q}_1 - \bar{q}_0) - \varepsilon_{01}^{(4)}(\nabla^2 \bar{q}_1 - \nabla^2 \bar{q}_0))\varphi_{01}\lambda_{01} \tag{4-3}$$

∇^2 denotes the undivided Laplacian operator

$$\nabla^2 \bar{q}_0 = \sum_{k=1}^{n_0} (\bar{q}_k - \bar{q}_0) = -n_0 \bar{q}_0 + \sum_{k=1}^{n_0} \bar{q}_k \,. \tag{4-4}$$

λ_{01} denotes the local spectral radius

$$\lambda_{01} = (|\bar{u}_{01} \cdot \bar{n}_{01}| + c_{01})S_{01} \tag{4-5}$$

where $\bar{u}_{01} = (\bar{u}_0 + \bar{u}_1)/2$ and $c_{01} = (c_0 + c_1)/2$ denote the cell face speed and cell face sound of speed respectively. $\bar{n}_{01}$ denotes the normal direction of the control surface to the edge between nodes v_0, v_1 and S_{01} its size.

A careful reader may observe that the formulation of the third differences in Equation (4-3) may lead to non-dissipative operator if there is a large variation in ε_{01}. Alternative definitions may have to be considered in the future.

φ_{01} is a factor introduced to account for the stretching in the grid. It is defined as

$$\varphi_{01} = 4\frac{\varphi_0\varphi_1}{\varphi_0 + \varphi_1} \tag{4-6}$$

where φ_0 is

$$\varphi_0 = \left\{\frac{\lambda_0}{4\lambda_{01}}\right\}^p. \tag{4-7}$$

the ratio between the integrated spectral radius where

$$\lambda_0 = \sum_{k=1}^{n_0} (|\bar{u}_{0k} \cdot \bar{n}_k| + c_{0k})S_k = \sum_{k=1}^{n_0} \lambda_{0k} \tag{4-8}$$

and where λ_{0k} is given in Equation (4-5). $p = 1/2$ is a factor that was chosen to have a close resemblance with the Martinelli eigenvalue scaling for structured grids [8]. This gives a dissipation proportional to the local spectral radius λ_{01} in the direction of the stretching. In the other directions a value slightly larger than the local spectral radius is obtained.

$\varepsilon_{01}^{(2)}$ is chosen to be active in the neighborhood of shocks and small in smooth regions of the flow:

$$\varepsilon_{01}^{(2)} = \kappa^{(2)}\left(\left|\sum_{k=1}^{n_0} (p_k - p_0)\right| \Big/ \sum_{k=1}^{n_0} (p_k + p_0)\right) sc_2 \tag{4-9}$$

where $\kappa^{(2)}$ is a constant coefficient provided as input and sc_2 is a scaling factor to reduce the dependency on the number of neighbors. The size of sc_2 is chosen such that for six neighbors the dissipation equals the dissipation in a structured scheme and a unit value is obtained,

$$sc_2 = \frac{12}{n_0 + n_1}. \tag{4-10}$$

The fourth difference dissipation is switched off in the vicinity of shocks:

$$\varepsilon_{01}^{(4)} = max(0, \kappa^{(4)} - \varepsilon_{01}^{(2)})\ sc_4 \tag{4-11}$$

where $\kappa^{(4)}$ is another user defined constant and sc_4 is a scaling factor chosen in accordance with sc_2:

$$sc_4 = \frac{36}{(n_0 + n_1)^2} \tag{4-12}$$

On coarser grids a simplified form of the artificial dissipation operator based on second differences only is used to save computational time but also to increase the amount of dissipation. The coarse grid operator looks like

$$\bar{d}_{01} = \varepsilon_{01}^{(0)}(\bar{q}_1 - \bar{q}_0)\varphi_{01}\lambda_{01} \tag{4-13}$$

where

$$\varepsilon_{01}^{(0)} = \kappa^{(0)} sc_2 \frac{ndim}{3} \tag{4-14}$$

and where $ndim$ is the dimension and where $\kappa^{(0)}$ is another user defined constant.

At a boundary the artificial dissipation should not contribute, i.e. the flux on the boundary is set to zero. In addition, a boundary condition on the second derivatives in the undivided Laplacian (4-4) is required. Following Mavriplis[7] the conservative variables are extrapolated linearly which corresponds to a normal second derivative,

$$\frac{\partial^2}{\partial n^2}\bar{q} = 0. \tag{4-15}$$

Requiring no normal derivative of the variables in the computation of the Laplacian is equivalent to only account for the contributions along the boundary. With the notation in Figure 3-2 the undivided Laplacian becomes

$$\nabla^2 \bar{q}_1 = \sum_{k = 2, 5} (\bar{q}_k - \bar{q}_1) = \bar{q}_2 - 2\bar{q}_1 + \bar{q}_5. \tag{4-16}$$

The pressure sensor in Equation (4-9) is not modified at a boundary, nor are the scaling factors sc_2 and sc_4. In Figure 3-2 there are four flux contributions to the

residual in node v_1, $\bar{d}_{12}$, $\bar{d}_{13}$, $\bar{d}_{14}$ and $\bar{d}_{15}$, the number of legs to node v_1 is $n_1 = 4$.

4.1.2 Upwind schemes

In addition to the central scheme, upwind schemes are available of second order accuracy. The upwind scheme is of Roe flux difference splitting type[9] as opposed to the more commonly used MUSCL type upwind schemes [10][11]. The main reason for this is to have a scheme as similar as possible to what is available in *EURANUS*, the structured counterpart [3]. In addition, the Roe scheme requires less memory than MUSCL type of scheme.

As for the central scheme, the convective term is computed as a central part with additional dissipation. The central part is computed as an average of the fluxes through:

$$\bar{f}_{01} = \frac{1}{2}(\bar{f}(\bar{q}_0) + \bar{f}(\bar{q}_1)) - \bar{d}_{01} \quad , \tag{4-17}$$

compare Equation (4-2).

The upwind dissipation $\bar{d}_{01}$ is here computed as

$$\bar{d}_{01} = \frac{1}{2}\tilde{R\Lambda R}^{-1}(\bar{q}_1 - \bar{q}_0) = \frac{1}{2}\tilde{R\Lambda L}^{-1}(\bar{v}_1 - \bar{v}_0) = \frac{1}{2}\tilde{R\Lambda} d\bar{w}_{01} \tag{4-18}$$

where $\bar{q}, \bar{v}$ denote the conservative and the primitive variables respectively, the primitive variables being the ones used in the computer code. $d\bar{w}_{01} = L^{-1}(\bar{v}_1 - \bar{v}_0) = R^{-1}(\bar{q}_1 - \bar{q}_0)$ denotes the characteristic variables. The tensor R is the right eigenvector matrix to the flux Jacobian,

$$\frac{\partial}{\partial \bar{q}}\bar{f} = R\Lambda R^{-1} \tag{4-19}$$

where the diagonal tensor Λ contains the eigenvalues. A similar expression can be obtained for the tensor L belonging to the primitive variables. All matrices are given in Appendix A.

The diagonal matrix Λ is obtained as

$$\Lambda = \left|\Lambda^*\right|(I - \Phi) \tag{4-20}$$

where Φ is a diagonal matrix with limiters for second order accuracy. Note that $\Phi = 0$ for a first order scheme.

For a Roe flux difference splitting scheme the components of R, L, Λ must be computed from the Roe averaged variables

$$\begin{aligned} \rho_{01} &= \sqrt{\rho_0}\sqrt{\rho_1} \\ \bar{u}_{01} &= \frac{\bar{u}_0\sqrt{\rho_0} + \bar{u}_1\sqrt{\rho_1}}{\sqrt{\rho_0} + \sqrt{\rho_1}} \\ H_{01} &= \frac{H_0\sqrt{\rho_0} + H_1\sqrt{\rho_1}}{\sqrt{\rho_0} + \sqrt{\rho_1}} \\ c_{01}^2 &= (\gamma - 1)\left[H_{01} - |\bar{u}_{01}|^2\right] \end{aligned} \quad . \tag{4-21}$$

However, arithmetic averages usually provide good solutions and are computationally less expensive and may be used as an option.

The diagonal matrix Λ^* in Equation (4-20) contains the eigenvalues adjusted with an entropy fix to prevent the eigenvalues to become zero and produce unphysical solutions. The following entropy fix is used for each of the eigenvalues:

$$|\lambda_i|^* = \begin{cases} \dfrac{\lambda_i^{\,2} + \delta^2}{2\delta} & ,|\lambda_i| \le \delta \\ |\lambda_i| & ,|\lambda_i| > \delta \end{cases} \tag{4-22}$$

where δ is a small fraction of the spectral radius, usually around 5%.

To achieve second order accuracy the limiter Φ in Equation (4-20) has to be computed from divided differences of the solution. Here we chose differences of the characteristics to avoid oscillations in the pressure [12], although it is computationally more expensive.

Gradients of all primitive variables are needed to compute the characteristics in the nodes. The gradient in a node is computed by evaluating the surface integral of the gradient theorem

$$\nabla v_0 = \frac{1}{V_0}\oint_{\partial V_0} v\bar{n}\,dS \cong \frac{1}{V_0}\sum_{k=1}^{n_0}\frac{1}{2}(v_k + v_0)\bar{n}_{0k}S_{0k} \tag{4-23}$$

where v denotes a component of the primitive variables. The node valued characteristics $d\bar{w}_0, d\bar{w}_1$ may be obtained as

$$d\bar{w}_0 = L_0(\nabla\bar{v}_0 \cdot (\bar{x}_1 - \bar{x}_0)),\ d\bar{w}_1 = L_1(\nabla\bar{v}_1 \cdot (\bar{x}_1 - \bar{x}_0)) \tag{4-24}$$

in addition to the face value $d\bar{w}_{01}$ in Equation (4-18).

However, on a structured Cartesian grid these characteristics correspond to a broad stencil, $d\bar{w}_i = (\bar{w}_{i+1} + \bar{w}_{i-1})/2$ using the expression in (4-18). To have a scheme that provides identical results compared to a structured scheme on a regular grid the following expression is used to compute the node valued characteristics:

$$\begin{aligned} d\bar{w}_0 &= 2L_0(\nabla\bar{v}_0 \cdot (\bar{x}_1 - \bar{x}_0)) - d\bar{w}_{01} \\ d\bar{w}_1 &= 2L_1(\nabla\bar{v}_1 \cdot (\bar{x}_1 - \bar{x}_0)) - d\bar{w}_{01} \end{aligned} \tag{4-25}$$

which, on a regular Cartesian structured grid corresponds to

$$d\bar{w}_{01} = d\bar{w}_{i+\frac{1}{2}} = \bar{w}_{i+1} - \bar{w}_i,\ d\bar{w}_0 = \bar{w}_i - \bar{w}_{i-1},\ d\bar{w}_1 = \bar{w}_{i+2} - \bar{w}_{i+1} \tag{4-26}$$

i.e. $d\bar{w}_0, d\bar{w}_1$ correspond to the left and right compact differences the for the cell face $i + \frac{1}{2}$ flux.

The limiter considered is the so called monmod limiter. The minmod function chooses the argument with the smallest amplitude provided all arguments have the same sign. If the signs are different the limiter is zero and the scheme reduces locally to a first order accurate one. A more compressive limiter may be considered in the future, the minmod limiter provides surprisingly good results though.

Two different ways of limiting is available. In the first way the limiter is computed as

$$\phi dw_{01} = minmod(dw_0, dw_{01}, dw_1) \tag{4-27}$$

where ϕ, dw denote a component of $\Phi, d\bar{w}$ respectively. This way of limiting is similar to the new family of symmetric TVD schemes used in *EURANUS* [3].

The second way of limiting is more consistent with the flux difference splitting technique. The limiter is then computed depending on the sign of the eigenvalues to the flux Jacobian.

$$\phi dw_{01} = \begin{cases} minmod(dw_0, dw_{01}) & ,\lambda_i \geq 0 \\ minmod(dw_{01}, dw_1) & ,\lambda_i < 0 \end{cases}. \tag{4-28}$$

This way of limiting is the choice for all upwind computations in this paper.

At boundaries no particular modification to the scheme is made. The numerical flux due to the upwind dissipation is zero.

The upwind dissipation requires about twice the amount of computational effort compared to the artificial dissipation for the central scheme. It is relatively cheap on memory, only the gradients of the primitive variables need to be stored.

4.2 Viscous terms

The viscous flux $\dot{f}_{V0k}$ in Equation (4-1) is computed in accordance with the convective flux. As seen from Equation (2-4) the viscous flux contains gradients that need to be computed in a way that leads to a compact discretization.

Below three different discretizations of the viscous terms are described. The first alternative, using gradients computed in the nodes, does not lead to a compact scheme and is therefore not evaluated in the computations. The two remaining alternatives are tested and evaluated numerically.

4.2.1 Using node gradients

There is a variety of different methods to choose from when making a discrete approximation of the viscous terms. Perhaps the most obvious method to approximate the gradients in the stress tensor (2-5), (2-15) is to use the Green-Gauss formulation to compute the gradients in the nodes:

$$\nabla\varphi_0 = \frac{1}{V_0}\oint_{\partial V_0}\varphi\bar{n}dS \cong \frac{1}{V_0}\sum_{k=1}^{n_0}\frac{1}{2}(\varphi_k+\varphi_0)\bar{n}_{0k}S_{0k} \tag{4-29}$$

where φ denotes some variables for which the gradient is required.

This is a natural way to compute the gradients since, with the edge based data structure, only node and edge data are available in the flow solver. The gradients are required at the control surface on the edge and are then obtained from the average of the two node values of the edge.

It is easy to realize that the average of nodal values of a gradient leads to a non-compact second difference. In one dimension on a regular structured grid this would imply a second derivative involving every second point:

$$\left(\frac{\partial^2\varphi}{\partial x^2}\right)_i \sim \frac{\varphi_{i+2}-2\varphi_i+\varphi_{i-2}}{4\Delta x^2} = \frac{\partial^2\varphi}{\partial x^2}+\frac{1}{3}\Delta x^2\varphi_{xxxx}+O(\Delta x^4) \tag{4-30}$$

This discretization does not damp the highest frequencies and can hence not prevent odd-even decoupling. Most likely though, the artificial dissipation added to the convective terms will stabilize the scheme but will add more diffusion as a consequence. More compact discretizations are considered below.

4.2.2 Correction to node gradients

Weiss *et al*. [13] propose a correction to obtain a more compact stencil involving all neighboring points. With reference to Figure 3-1 on page 13 the averaged gradient on the edge between node ν_0 and node ν_1 can be corrected as:

$$\nabla\varphi'_{01} = \nabla\varphi_{01} + \frac{1}{|\bar{x}_1 - \bar{x}_0|^2}(\delta\varphi_{01} - (\nabla\varphi \cdot (\bar{x}_1 - \bar{x}_0)))(\bar{x}_1 - \bar{x}_0) \tag{4-31}$$

where $\nabla\varphi_{01}$ is the averaged, non-corrected gradient

$$\nabla\varphi_{01} = \frac{1}{2}(\nabla\varphi_0 + \nabla\varphi_1) \tag{4-32}$$

and where $\nabla\varphi_0$, $\nabla\varphi_1$ are computed with the Green-Gauss formulation as in Equation (4-29). $\delta\varphi_{01}$ is defined as

$$\delta\varphi_{01} = \varphi_1 - \varphi_0 . \tag{4-33}$$

This is a simple correction to do for the gradients, it is also easy to show that it leads to compact second differences on a regular grid. Such a discretization in one dimension will lead to:

$$\left(\frac{\partial^2\varphi}{\partial x^2}\right)_i \sim \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{\Delta x^2} = \frac{\partial^2\varphi}{\partial x^2} + \frac{1}{12}\Delta x^2\varphi_{xxxx} + O(\Delta x^4) \tag{4-34}$$

Not only does this discretization damp the highest frequencies, in addition the truncation error is less than with the average of nodal gradients leading to the discretization in Equation (4-30).

4.2.3 Thin-layer approximation

Another approach to obtain a compact viscous stencil is to divide the viscous operator in a part with normal derivatives and remaining tangential derivatives. If only the normal parts of the viscous operator is included in the discretization of the Navier-Stokes equations, the governing equations are said to be a thin-layer approximation.

The viscous flux for the momentum equations can be divided as

$$\tau_{ij}n_j = (\tau_{ij}n_j)_{tl} + (\tau_{ij}n_j)_{\tan} \tag{4-35}$$

where $(\tau_{ij}n_j)_{tl}$ contains only normal derivatives and leads to a thin-layer discretization if only this term is considered. A fully viscous approximation is obtained if also the remaining part of the viscous terms $(\tau_{ij}n_j)_{\tan}$ is added. $\bar{n} = (n_x, n_y, n_z) = (n_1, n_2, n_3)$ are the components of the normal.

The thin-layer part can then be formulated as [14]

$$(\tau_{ij}n_j)_{tl} = \mu\left(\frac{\partial u_i}{\partial n} + \frac{1}{3}\left(\frac{\partial u_j}{\partial n}n_j\right)n_i\right) \tag{4-36}$$

The normal derivatives in Equation (4-36) can be approximated on the edges as

$$\frac{\partial \varphi_{01}}{\partial n} = \frac{\varphi_1 - \varphi_0}{|\bar{x}_1 - \bar{x}_0|} \tag{4-37}$$

with the notation from Figure 3-1 and where the normal is directed from node v_0 to node v_1. With this formulation only two points are involved in computing the normal gradients at the edges and hence automatically leads to a compact second derivative. On a regular grid in one dimension a compact second difference is obtained as in Equation (4-34).

By recalling the identity of the Laplace's equation

$$\int_\Omega \Delta\phi dV = \oint_{\partial\Omega} \frac{\partial \phi}{\partial n} dS \tag{4-38}$$

the following approximation of the Laplace's equation at node v_0 is obtained:

$$\Delta\varphi_0 \approx \frac{1}{V_0}\sum_{k=1}^{n_0} \frac{S_{0k}}{|\bar{x}_1 - \bar{x}_0|}(\varphi_k - \varphi_0) \tag{4-39}$$

It leads to a positive discretization of the Laplace's equation since the maximum principle associated with Laplace's equation requires the coefficients $\frac{S_{0k}}{|\bar{x}_1 - \bar{x}_0|}$ to be positive which is satisfied. Haaselbacher [15] points out that there is a basic incompatibility between accuracy and positivity of the coefficients on general grids. Second order accuracy and positivity can only be achieved simultaneously on regular grids [16].

The remaining parts of the viscous terms $(\tau_{ij}n_j)_{\tan}$ contain gradients which may be added using the Green-Gauss formulation in Equation (4-29) and thus a fully viscous approach can be obtained.

4.3 Source terms

The source term $S_{t\,0}$ in Equation (4-1) is computed directly at node v_0. The source term contains gradients according to Equations (2-12) and (2-13) that are approximated with the Green-Gauss formulation in Equation (4-29).

5 Time Integration

The governing equations may either be integrated to steady state or in a time accurate fashion. Steady state calculations use an explicit Runge-Kutta scheme to integrate to steady state. The Runge-Kutta coefficients are defined by the user input, usually a first order accurate scheme is chosen with good high-frequency damping. Steady state calculations use a local time step to speed up the convergence, implicit residual smoothing is also available. Only steady state calculations are described here.

5.1 Explicit Runge-Kutta

To integrate (4-1) to steady state a fully explicit Runge-Kutta time stepping scheme is used. An explicit m-stage Runge-Kutta scheme for the equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\bar{q} - \bar{R}(\bar{q}) = 0 \tag{5-1}$$

can be written

$$\begin{aligned} \bar{q}^1 &= \bar{q}^n - \alpha_1 \Delta t \bar{R}(\bar{q}^n) \\ \bar{q}^2 &= \bar{q}^n - \alpha_2 \Delta t \bar{R}(\bar{q}^2) \\ &\cdots \\ \bar{q}^m &= \bar{q}^n - \alpha_m \Delta t \bar{R}(\bar{q}^{m-1}) \\ \bar{q}^{n+1} &= \bar{q}^m \end{aligned} \tag{5-2}$$

For steady state calculations a 3-stage, first order accurate scheme which provides good smoothing for both central and upwind schemes is recommended,

$$\alpha_1 = 0.66667,\ \alpha_2 = 0.66667,\ \alpha_3 = 1.0\ . \tag{5-3}$$

This scheme allows good smoothing properties at $CFL = 1.5$ with one computation of the artificial viscosity in the first stage. Its stability region in the complex plane can bee seen in Figure 5-1.

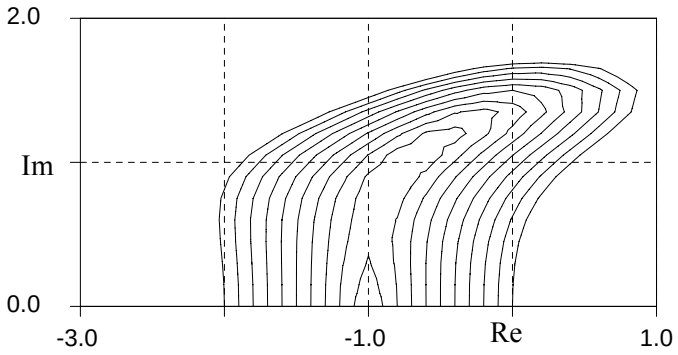


Fig. 5-1 Stability region in the complex plane of a three-stage Runge-Kutta scheme with one evaluation of the viscous terms.

The Runge-Kutta coefficients are defined by the user input, so is also the choice of in which stages to compute the dissipative terms.

5.2 Local time step

The local time step is computed for each node v_0 according to

$$\Delta t_0 = min\left(CFL\frac{V_0}{\lambda_0}, CFLVIS\frac{V_0}{\lambda_{V0}}\right) \tag{5-4}$$

where V_0 is the volume and λ_0 is the integrated convective spectral radius according to Equation (4-8). λ_{V0} is the corresponding viscous spectral radius

$$\lambda_{V0} = \sum_{k=1}^{n_0} \tilde{\mu}_{0k} 2^{ndim} \frac{S_k^2}{\rho_{0k}} \tag{5-5}$$

where μ_{0k} is the sum of the dynamic and turbulent viscosity in a turbulent calculation on the edge between nodes v_0 and v_k, i.e. $\tilde{\mu}_{0k} = 1/2(\tilde{\mu}_0 + \tilde{\mu}_k)$. CFL and $CFLVIS$ are user input and chosen according the stability region of the Runge-Kutta scheme used.

5.3 Implicit residual smoothing

To increase the maximum time step, implicit residual smoothing is employed[6]. The smoothing is employed for each residual to all equations independently. The smoothing for node v_0 can for the residual of each unknown be written

$$\hat{R}_0 = R_0 + \varepsilon \nabla^2 \hat{R}_0 \tag{5-6}$$

where ε is a constant, ∇^2 is the undivided Laplacian defined in Equation (4-4). Here $\hat{R}_0$ indicates a new smoothed residual. This may be written

$$(1 + n_0\varepsilon)\hat{R}_0 - \varepsilon \sum_{k=1}^{n_0} \hat{R}_k = R_0 \tag{5-7}$$

For a structured solver a tridiagonal system of equations is obtained. For the unstructured solver a sparse, diagonally dominant matrix is obtained. The sparse matrix is not inverted exactly, instead a few Jacobi iterations of Equation (5-7) are done. This results in the following iterative scheme for the residual smoothing:

$$R_0^{n+1} = \frac{R_0^0 + \varepsilon \sum_{k=1}^{\upsilon} R_k^n}{1 + \varepsilon n_0}; \quad n \geq 0 \tag{5-8}$$

Usually 2 iterations gives sufficient smoothing.

Theoretically for structured grids where implicit residual smoothing is combined with the direct solution of tridiagonal systems, the time step can be increased according to the value of ε,

$$\alpha = \frac{CFL^*}{CFL} \leq \sqrt{4\varepsilon + 1} \tag{5-9}$$

where CFL, CFL^* are the CFL numbers of the unsmoothed and smoothed scheme. Typical values using a structured grid is $\alpha = 2$ corresponding a doubling of the CFL number and to $\varepsilon = 0.75$.

Here, where a few Jacobi iterations are carried out, usually

$$\alpha = \frac{CFL^*}{CFL} \leq 1.3 . \tag{5-10}$$

for good efficiency. This correspond to a value of approximately $\varepsilon \leq 0.2$.

On stretched grids only smoothing in the direction of the stretching should be used. The smoothing in the other directions must be reduced or removed. With structured grids this is achieved by letting the coefficient ε be a function of geometric quantities, see e.g. Radespiel *et al.* [17]. A similar procedure is used here.

There are several ways to modify the Equation (5-8) to account for the stretching in the grid. Currently two ways are implemented. One way to account for the stretching is defined as:

$$R_0^{n+1} = \frac{R_0^0 + \varepsilon n_0 \sum_{k=1}^{n_0} R_k^n \Psi_k / \sum_{k=1}^{n_0} \Psi_k}{1 + \varepsilon n_0}; \quad n \geq 0 \tag{5-11}$$

where Ψ_k contains the geometric information

$$\Psi_k = \frac{S_{0k}^2}{|x_k - x_0|} \tag{5-12}$$

This type of smoothing increases slightly the smoothing in the direction of the stretching when it is removed in the other directions. On a regular grid it reduces to Equation (5-8). This smoothing is also applied to the corrections in the multi-grid procedure.

The other way to smooth is based on a procedure from Mavripilis[18] where Equation (5-8) is modified as:

$$R_0^{n+1} = \frac{R_0^0 + \sum_{k=1}^{n_0} \varepsilon_{0k} R_k^n}{1 + \sum_{k=1}^{n_0} \varepsilon_{0k}}; \quad n \geq 0 \tag{5-13}$$

where

$$\varepsilon_{0k} = max\left(\frac{1}{4}((\alpha\varphi_{0k})^2 - 1), 0\right) \tag{5-14}$$

and where φ_{0k} is a function to account for the local stretching in the grid and is defined as

$$\varphi_{0k} = s\frac{1 + s^{-p}}{1 + s} \tag{5-15}$$

and where s is another stretching factor defined as

$$s = \frac{S_{0k}}{max S_{0i}}. \tag{5-16}$$

where index i denotes all indices of nodes connected to node v_0.

6 Boundary Treatment

Implemented boundary conditions in the *EDGE* code are:

- Euler wall,
- Symmetry plane,
- Viscous wall, adiabatic or isothermal,
- Farfield boundary using characteristic boundary conditions.

The Euler wall, symmetry plane and the farfield boundary conditions use a weak formulation, i.e. the boundary conditions are imposed through the flux and all unknowns on these boundaries are updated like any interior unknown.

On viscous walls, however, the velocity is imposed strongly through a no-slip condition. In addition, the density is imposed strongly for an isothermal wall. A strong formulation implies that values of a strongly imposed variable on the boundary are explicitly fixed, i.e. they remain at their imposed values and are not considered as unknowns.

The notations from Figure 3-2 are used in the following. Note that only the central, inviscid terms and the viscous terms that contribute to the fluxes on the boundaries, the dissipative fluxes are set to zero.

6.1 Euler wall and symmetry plane

At an Euler wall the normal component of the velocity is zero

$$\bar{u}_1 \cdot \bar{n} = 0 \tag{6-1}$$

and hence the inviscid wall flux becomes

$$\vec{f}_1 = \begin{pmatrix} 0 \\ p_1 n_x S \\ p_1 n_y S \\ p_1 n_z S \\ 0 \end{pmatrix} . \tag{6-2}$$

The fluxes are added to the residuals for node ν_1. The same boundary condition is used for a symmetry plane. Note that condition (6-1) is only implied through the flux, the unknown velocity itself will not necessarily satisfy this condition exactly. This is a consequence of using a weak boundary condition.

6.2 Viscous wall

On a viscous wall the velocity is imposed strongly through a no-slip boundary condition

$$\bar{u}_1 = 0 \tag{6-3}$$

Both a weak and a strong formulation have been tested and evaluated. The strong formulation has shown better rates of convergence and has therefore been the chosen one. With a strong formulation, the residual of the velocity on the boundary does not need to be solved for since the velocity will be kept constant according to Equation (6-3). This also implies that the fluxes for the velocity need not to be computed. In addition to the velocity, for a turbulent calculation, also the turbulent quantities are imposed strongly through

$$k_1 = 0 \tag{6-4}$$

$$\omega_1 = \frac{9\mu_1}{\beta\rho_1|\bar{x}_1 - \bar{x}_3|^2} \tag{6-5}$$

where the value ω_1 on the boundary is obtained as for structured grids recommended by Hellsten [19], β is a constant $\beta = 0.075$ and $|\bar{x}_1 - \bar{x}_3|$ is the distance from the wall node ν_1 to the closest internal node (ν_3) for which $|\bar{x}_1 - \bar{x}_k|$ forms an angle to the boundary that is closest to orthogonal.

At an iso-thermal wall there is contribution from the viscous terms to the energy equation in Equation (2-4). The remaining boundary flux becomes

$$f_{E_1} = -\kappa\frac{\partial T}{\partial n_1} \tag{6-6}$$

where the gradient on the boundary node is computed as the difference of the temperature in the interior node and the wall node

$$\frac{\partial T}{\partial n_1} = \frac{T_3 - T_1}{|\bar{x}_1 - \bar{x}_3|} \tag{6-7}$$

For turbulent calculations there is an additional contribution from the turbulence in Equations (2-11), (2-16) and the corresponding boundary flux becomes

$$f_{E_1} = -(\kappa + \kappa_T)\frac{\partial T}{\partial n_1} \tag{6-8}$$

The wall density is imposed strongly at an isothermal wall through the relation

$$\rho_1 = \frac{p_1}{RT_w} = \frac{(\gamma - 1)E_1}{RT_w} \tag{6-9}$$

where T_w is the specified wall temperature.

At an adiabatic wall there is no contribution from the viscous terms to the energy equation at a wall since the temperature gradient is zero,

$$\frac{\partial T}{\partial n_1} = 0 \tag{6-10}$$

and hence the boundary flux is zero for both the density and energy equation.

6.3 Farfield boundary using characteristics

Characteristic boundary conditions are used at the farfield. These boundary conditions can be used for both subsonic and supersonic in- and outflow where the characteristics are either set from free stream quantities for ingoing characteristics or extrapolated from the interior for outgoing characteristics.

Primitive variables are used and stored in the program since they lead to sparse and computationally less expensive expressions.

Given a set of local primitive variables $\bar{v}_1 = (\rho, u, v, w, p)^T$ at the boundary a new set of primitive variables $\bar{v}_1'$ have to be computed to be used in the flux evaluation for node v_1. The characteristic variables are denoted $\bar{w}$ and the relation between the primitive and characteristic variables is

$$\bar{v} = L\,\bar{w} \tag{6-11}$$

where L is given in appendix A with its inverse.

By computing characteristics based on both local and free stream primitive variables,

$$\bar{w}_L = L^{-1}\,\bar{v}_L, \quad \bar{w}_\infty = L^{-1}\,\bar{v}_\infty \tag{6-12}$$

either local or the free stream characteristics are used depending on the sign of the eigenvalue. The local variables $\bar{v}_L$ are computed by local extrapolation,

$$\bar{v}_L = \bar{v}_1 . \tag{6-13}$$

and the components of the transformation matrices L, L^{-1} are computed using free stream values and the local surface normal vector,

$$L = L(\bar{v}_\infty, \bar{n}), \quad L^{-1} = L^{-1}(\bar{v}_\infty, \bar{n}) \tag{6-14}$$

Depending on the sign of the eigenvalue the components of characteristics $\bar{w}_1$ can be obtained

$$w_{1i} = \begin{cases} w_{Li} & , \ \lambda_i > 0 \\ w_{\infty i} & , \ \lambda_i \leq 0 \end{cases} \tag{6-15}$$

where λ_i denotes the *i*th eigenvalue belonging to the *i*th characteristic. The variables $\bar{v}_1$ can then be obtained from (6-11) from which the flux on the boundary can be computed.

7 Numerical application

In this section four applications using the flow solver *EDGE* for external flow are presented.

The first one is an Euler case with subsonic flow over a RAE2822 airfoil where it is demonstrated that second order spatial accuracy can be obtained. The second case concerns viscous flow at a low Reynolds number over a NACA0012 airfoil with no addition of artificial dissipation. Different numerical discretizations of the viscous terms are studied for this case. The third case is viscous, turbulent flow over the RAE2822 airfoil. In addition, 3D flow over the ONERA M6 wing has been computed. In all cases, comparisons are made with results with *EURANUS,* a code for flow computations on structured grids, and to experimental data.

7.1 Subsonic Euler flow

The Euler part has been validated earlier [1]. An additional subsonic Euler flow has been computed to compare the solutions and especially the level of accuracy between the cell centered and node vertex approach. The level of accuracy is studied by comparing the maximum total pressure loss which occurs close to the leading edge. For a subsonic case, the pressure loss should approach zero as the grid is refined.

The RAE2822 airfoil was chosen since it is an airfoil often used in the validation of flow solvers. The conditions chosen are

$$M = 0.5, \alpha = 2.8^\circ . \tag{7-1}$$

A fine grid of 545×97 nodes was used from which a medium and a coarser grid were created by removing every second point in the two directions. The grid sizes can be seen in Tab. 7-1. These grids were also triangulated in order to see the influence from a different cell type. The grids are of C-type and the coarse grid with quadrilaterals and triangles can be seen in Figure 7-1.

Only fourth order damping is used, the value of the fourth order damping parameter (denoted VIS4 in the input to the solver) is $\kappa^{(4)} = 0.05$ which is a default value. A corresponding value is used in the *EURANUS* computations.

Tab. 7-1. Grid size of the three quadrilateral grids of the RAE airfoil

Grid	Size
Fine	545×97
Medium	273×49

Grid	Size
Coarse	137×25

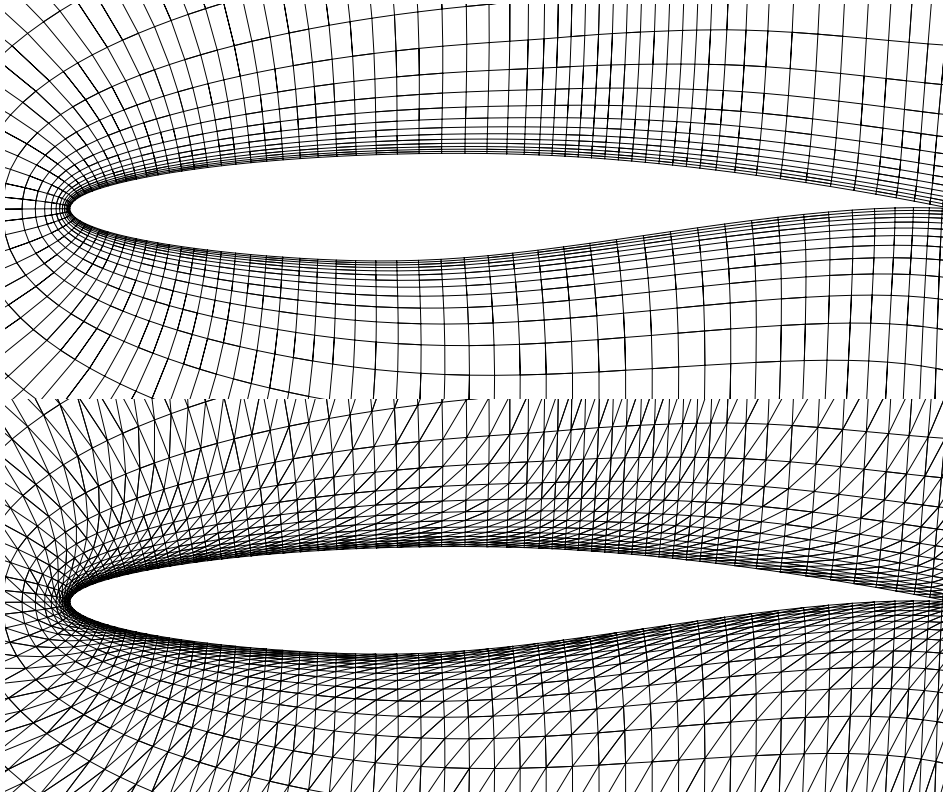


Fig. 7-1 The coarse grid for the inviscid computations over the RAE airfoil with 137×25 nodes. Quadrilaterals (top) and triangles (bottom).

The rate of convergence can be observed in Figure 7-2. The CFL number is CFL=1.5 and 5 multigrid levels were used. However, for the triangular grid it was only possible to use 2 multigrid levels to avoid divergence which explains the slower convergence. With the same grid and multigrid levels, the rate of convergence is similar between the cell centered structured and cell vertex unstructured approaches.

The pressure distribution of the three quadrilateral grids can be seen in Figure 7-3 and Figure 7-4. Very small differences can be observed.

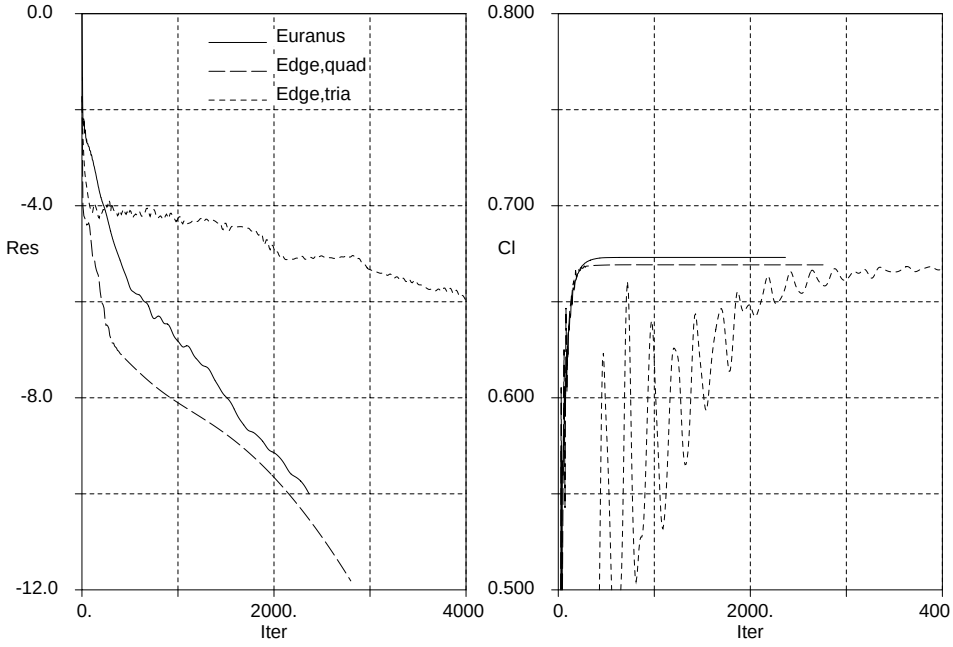


Fig. 7-2 Convergence of density residual and lift force. *EURANUS* solid line, *EDGE* quadrilateral grid long dash and *EDGE* triangular grid short dash. Only 2 multigrid levels with the triangular grid.

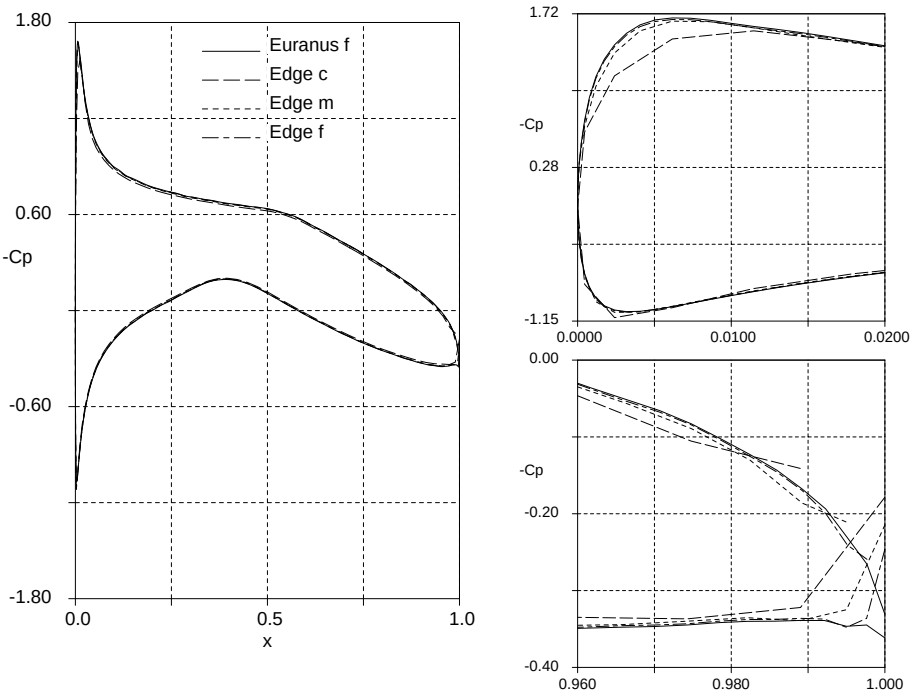


Fig. 7-3 Pressure distribution with the three quadrilateral grids compared to structured grid results on the finest grid. Upper right: blow up at leading edge, lower right: blow up at trailing edge. Structured grid results solid line, unstructured coarse grid results long dashes, medium grid results short dashes fine grid results log/short dashes.

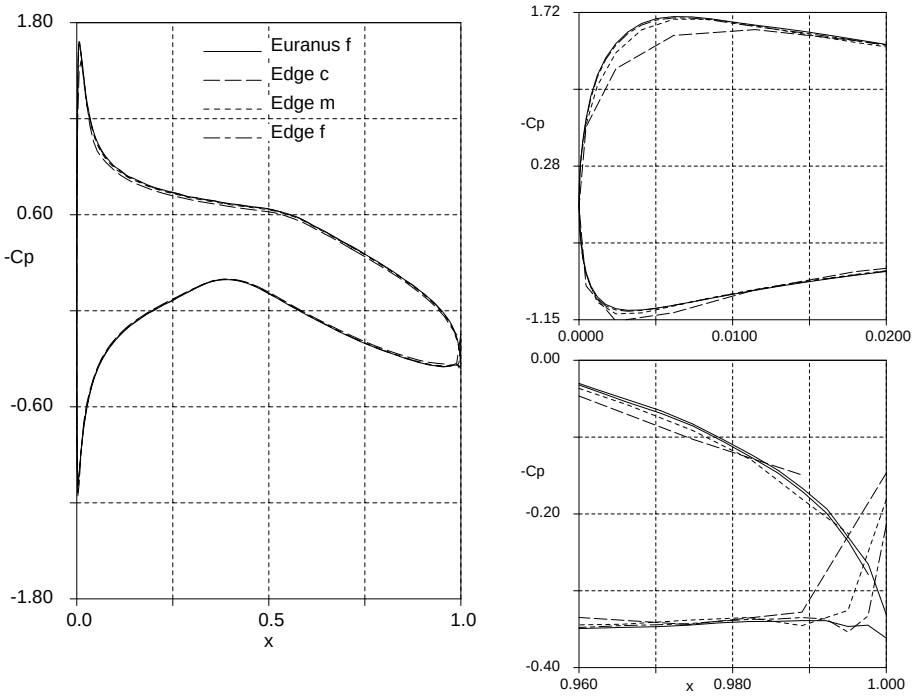


Fig. 7-4 Pressure distribution with the three triangular grids compared to structured grid results on the finest grid. Upper right: blow up at leading edge, lower right: blow up at trailing edge. Structured grid results solid line, unstructured coarse grid results long dashes, medium grid results short dashes fine grid results long/short dashes

The total pressure loss can be seen in Figure 7-5 for the cell vertex approach using the two cell types as well as for the cell centered approach. The loss is largest close to the leading edge except with the triangular grid where the largest losses occur at the trailing edge.

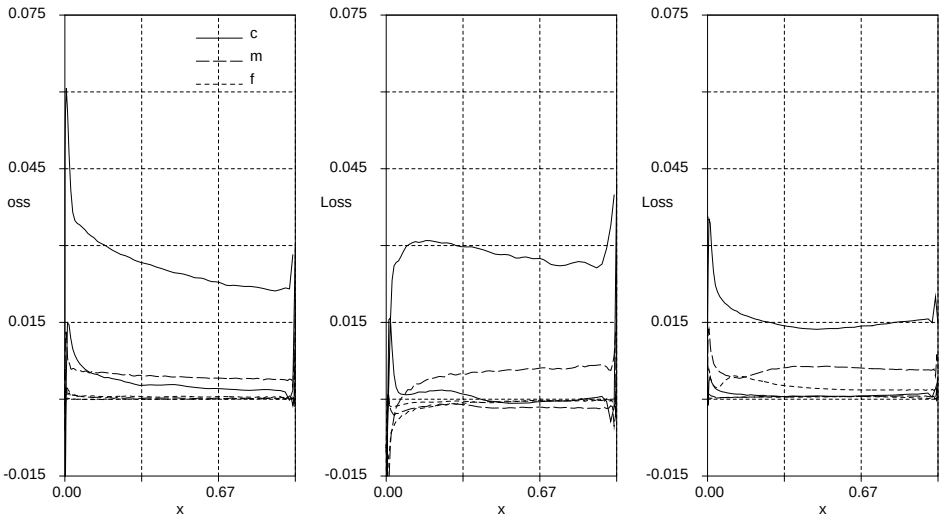


Fig. 7-5 Total pressure loss on coarse, medium and fine grid. Left: quadrilateral cell vertex (unstructured grid), middle: triangular, right: quadrilateral cell centre (structured grid). Coarse grid solid line, medium grid long dashes, fine grid short dashes.

The cell vertex approach for the quadrilateral grid gives a higher pressure loss on coarser grids compared to the cell centered approach. The reduction of the loss is increased as the grid is refined. This is also obvious in Figure 7-6 where the max-

imum total pressure loss is plotted as function of the grid size. The last 5% of the airfoil has not been taken into account due to the unphysical pressure losses that occur at the trailing edge. In this plot also results using the upwind scheme are plotted. All of these schemes are formally second order accurate in space and hence the maximum total pressure loss is expected to be reduced accordingly. In the figure the slope of a first and second order decay is plotted, the two lines on top.

It can be noted that the decay of the total pressure loss is close to second order for the node vertex (*EDGE*) central scheme with a quadrilateral grid and a somewhat higher decay with the triangular grid. The upwind schemes with the node vertex scheme decay with an order of accuracy slightly less than two. It is interesting to note that the decay is only first order accurate for the cell centered structured (*EURANUS*) approach.

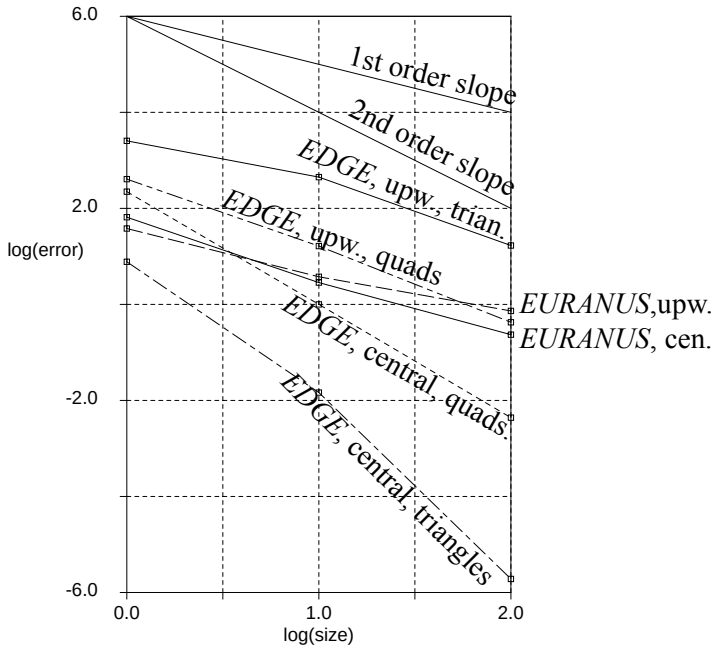


Fig. 7-6 Decay of maximum total pressure loss for the RAE2822 airfoils using as the grid is refined. The central and upwind schemes, cell centre (*EURANUS*) and cell vertex (*EDGE*) approaches and quadrilateral and triangular cell types are used.

7.2 Viscous flow over NACA0012

The flow over the NACA0012 airfoil at a Reynolds number of $Re = 500$ is used as the first case for the viscous extension of *EDGE*. This case has been used for validation earlier in a workshop [19]. One of the flow conditions from that workshop was chosen here:

$$M_\infty = 0.8, \alpha = 10°, Re = 500 \tag{7-2}$$

where the Reynolds number is based on the chord of the airfoil. A structured grid with 257×65 points was used. The structured grid was transformed into the

unstructured format required by *EDGE*. In addition, this grid has been triangulated. The grid with the two cell types can be seen in Figure 7-7.

The flow solution and the rate of convergence have been compared to results using the solver *EURANUS* for structured grids. The structured results are also obtained without use of artificial dissipation. In order to compare the rate of convergence, the same CFL numbers have been used for the structured and unstructured computations, $CFL = 1.5$ with three multigrid levels. Residual smoothing is applied for the computations with the unstructured approach, no implicit residual smoothing is applied in the structured approach due to the relatively low CFL number.

The two compact viscous operators Section 4.2 on page 21 are evaluated and a comparison between the fully viscous and thin layer approaches is carried out. Two different wall boundary conditions are validated, isothermal and adiabatic wall boundary conditions.

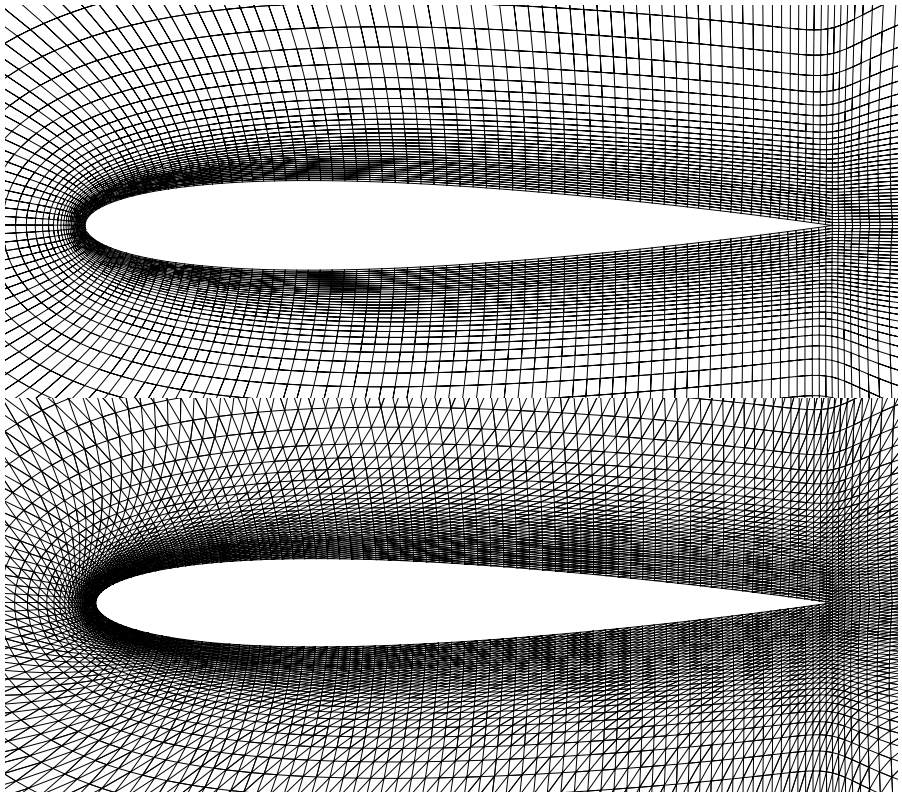


Fig. 7-7 The NACA0012 257×65 grid with quadrilaterals and triangles.

7.2.1 Isothermal wall

The wall temperature is kept constant to

$$T_{wall} = 338.4 \tag{7-3}$$

which is the stagnation temperature with a free stream temperature of $T_\infty = 300$.

Initially results using the grid with quadrilaterals is shown. The rate of converge for the density residual and the integrated lift force can be seen in Figure 7-8. Results are displayed using *EURANUS* with a fully viscous operator, the thin-layer approach described in "Thin-layer approximation" on page 22 (denoted t.l.), the same approach with the addition of remaining tangential components of the viscous operator to obtain a fully viscous operator (denoted f.n.) and finally results using a fully viscous operator with a correction to the gradients described in "Correction to node gradients" on page 22 (denoted GRC).

The convergence is considerably faster using a cell centered structured approach with *EURANUS* whereas it is about the same for different unstructured approaches. The reason behind this is unclear. The lowest lift is obtained using cell centered structured approach with *EURANUS*. Note also that with a fully viscous operator with corrected gradients (GRC) gives a slightly lower lift that with the two other discretizations of the viscous operator.

The pressure distribution for the fully viscous approach and the thin layer approach can be seen in Figure 7-9. In general, the differences are small. The pressure distribution with the cell centered structured approach gives a slightly lower stagnation pressure which agree better with the theoretical stagnation pressure given in Table 7-2 on page 40 and also explains the lower predicted lift force. All computations predict a too high stagnation pressure.

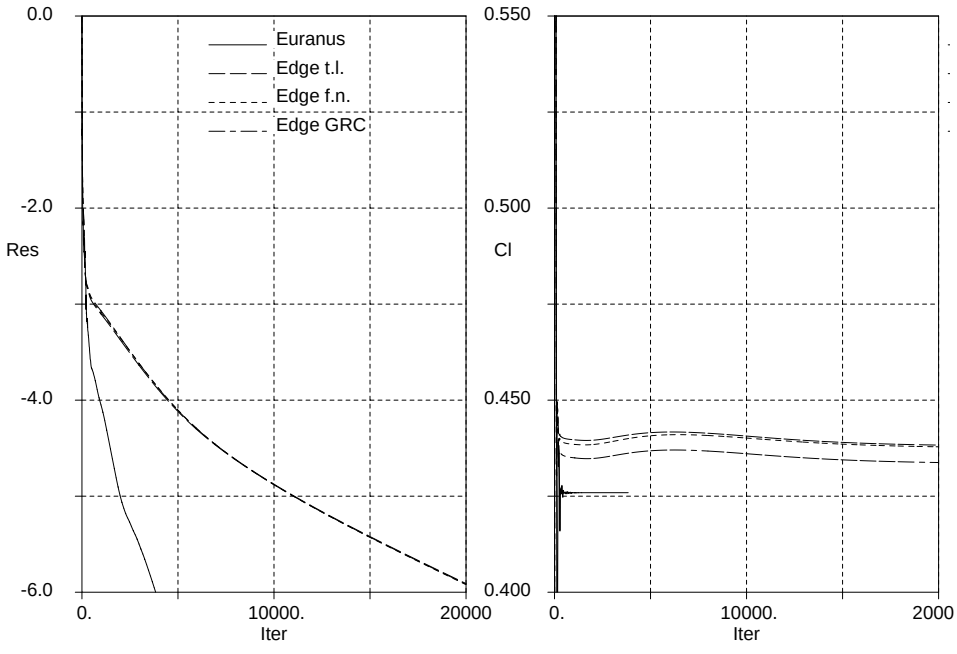


Fig. 7-8 Convergence of density residual and lift force. *EURANUS* fully viscous operator solid line, *EDGE* thin layer long dashes, *EDGE* fully viscous operator short dashes, *EDGE* fully viscous operator with mod. gradients long-short dashes.

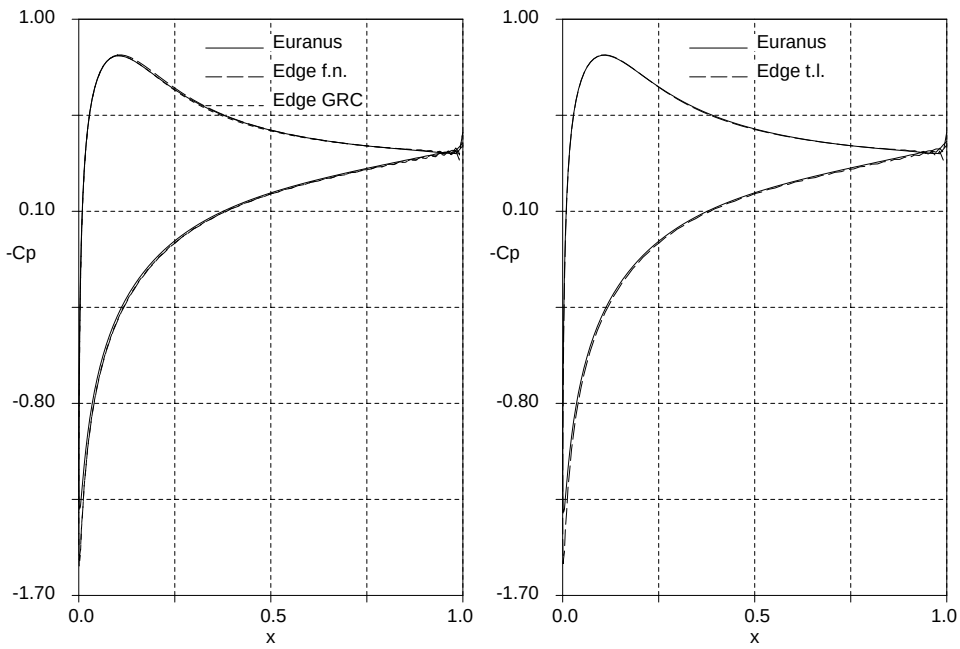


Fig. 7-9 Cp distribution. Left fully viscous operator, right thin-layer approach. *EURANUS* solid line, *EDGE* long dashes, *EDGE* with mod. gradients short dashes.

Tab. 7-2. Stagnation Cp for the different computations.

Theoretical value at stagnation	*EURANUS* thin-layer	Euranus fully viscous	*EDGE* thin-layer	*EDGE* full viscous	*EDGE* full viscous, corrected
1.17	1.31	1.29	1.55	1.53	1.55

Corresponding plots for skin friction and heat transfer can be seen in Figure 7-10 and Figure 7-11. Very small differences between all methods are observed for the skin friction. For the heat transfer the results using the structured and unstructured approaches agree well. One can notice the difference between the fully viscous approach and the thin-layer approach.

The skin friction reveals that there is a large separated region on the upper part of the airfoil. This can be seen in Figure 7-12 where iso Mach contours are plotted.

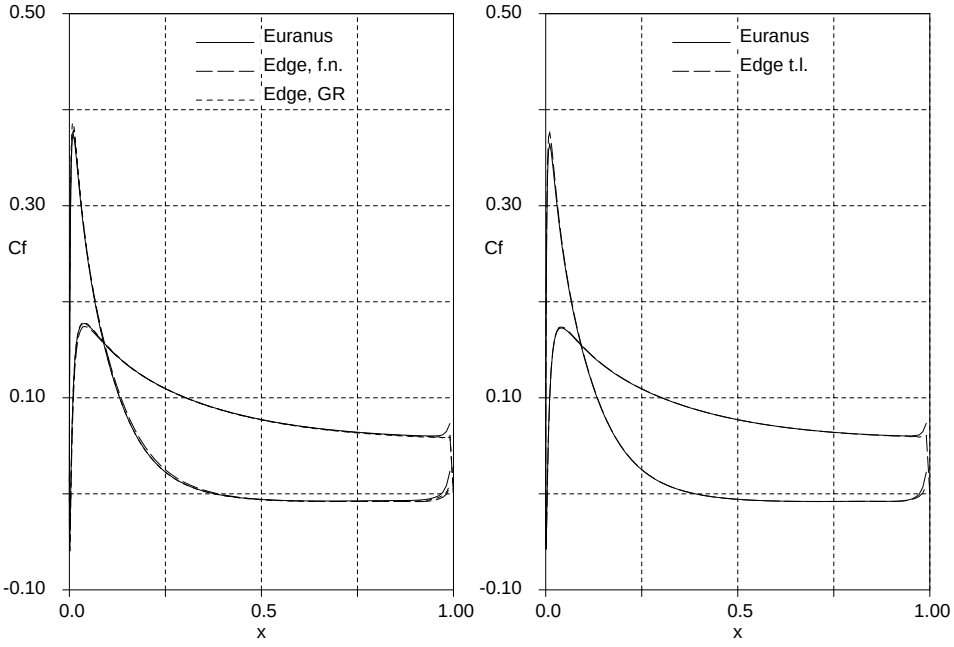


Fig. 7-10 Skin friction distribution. Left fully viscous operator, right thin-layer approach. *EURANUS* solid line, *EDGE* long dashes, *EDGE* with mod. gradients short dashes.

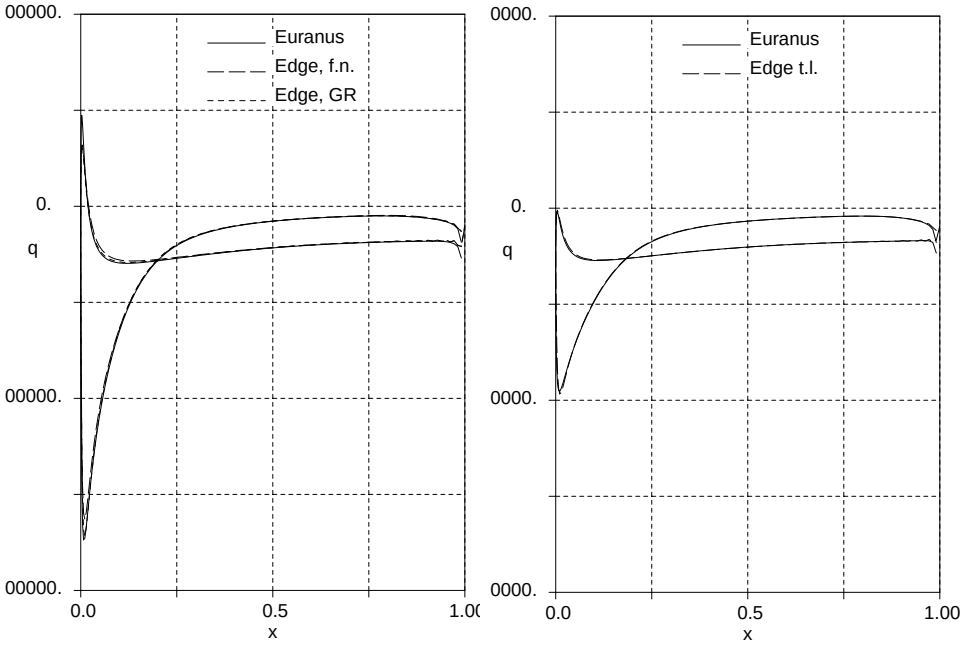


Fig. 7-11 Heat transfer distribution. Left fully viscous operator, right thin-layer approach. *EURANUS* solid line, *EDGE* long dashes, *EDGE* with mod. gradients short dashes

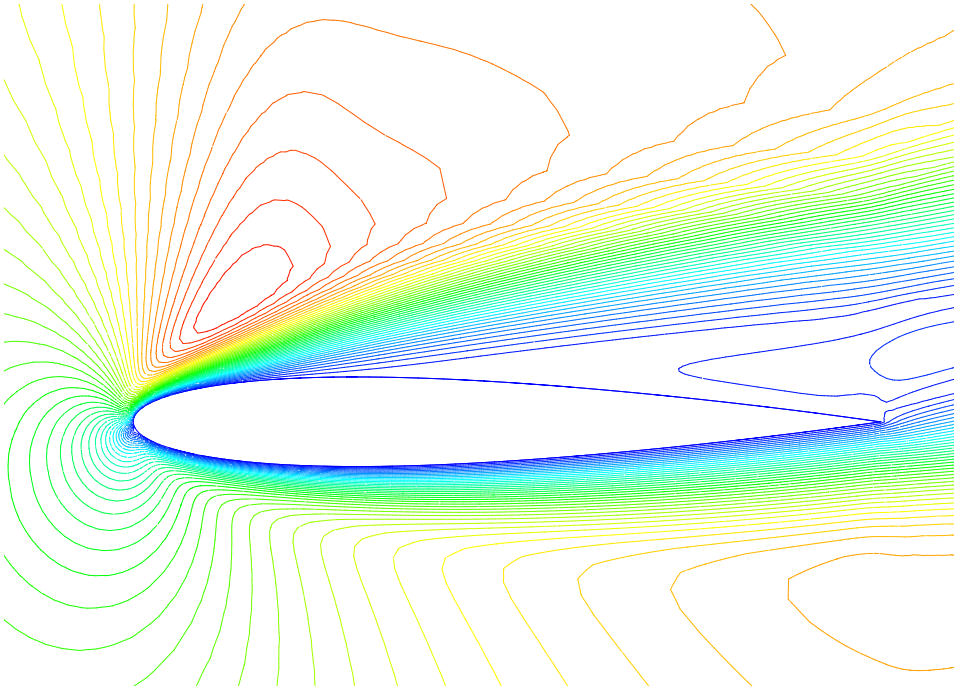


Fig. 7-12 Iso Mach contours for with the thin-layer, unstructured approach.

The skin friction and heat transfer distribution using the two different viscous discretization techniques terms in Section 4.2.2 on page 22 (solid line) and Section 4.2.3 on page 22 (dashed line) of the fully viscous operator are displayed in Figure 7-13. For the thin-layer approach in Section 4.2.3 on page 22, the remaining tangential part is computed with gradients using the Green-Gauss approach in Eq. 4-29, on page 21. There are very small differences between the two approaches, the rate of convergence is also practically unaffected.

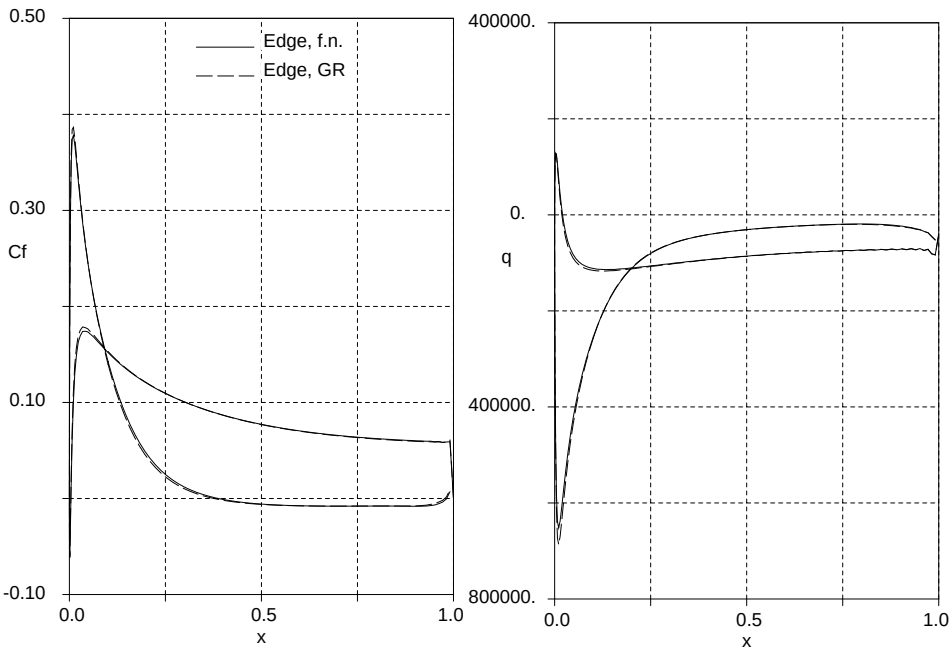


Fig. 7-13 Skin friction distribution (left) and heat transfer distribution (right) with fully viscous operator. Two different discretizations of the viscous terms.

The differences between the solution using the cell centered structured approach and the cell vertex unstructured approach seem to be small from the comparison above except for the rate of convergence and level of stagnation pressure. There are some differences though, in particular in the nose region just outside of the wall. This can be seen in the pressure distribution in Figure 7-14 at y=0 where there are oscillations with the node vertex approach growing in amplitude as the airfoil is approached. Using the cell centered approach there are practically no oscillations.

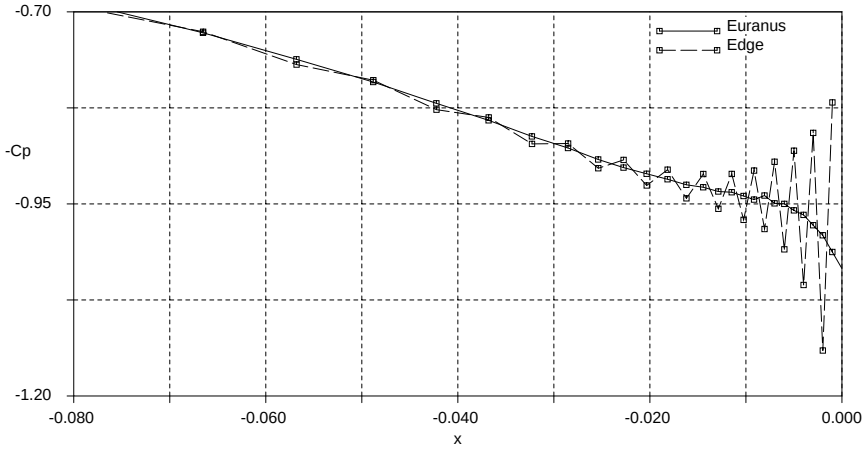


Fig. 7-14 Pressure distribution left of the nose region at y=0. Fully viscous discretization. Unstructured node vertex approach (solid line) and cell centre structured approach are compared (dashed line).

Results with the unstructured node vertex approach using the same but triangulated grid are displayed in Figure 7-15 and Figure 7-16. The impact on the rate of convergence is small but there clear differences in the solutions, most obvious in the heat transfer. The pressure distribution reveals oscillations with the triangular grid. A possible explanation to the differences is how the dual grid is constructed. The present approach of using the midpoints of the cells to construct the dual grid leads to rather skewed dual grids (i.e. control volumes) as the triangles are stretched close to the airfoil.

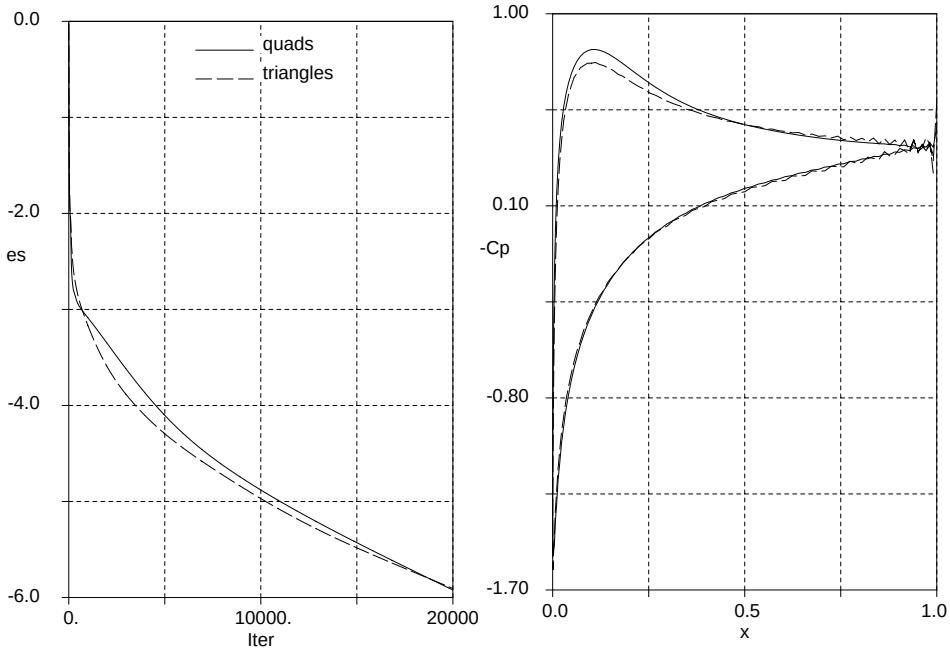


Fig. 7-15 Convergence of density residual (left) and Cp distribution full Navier-Stokes (right) with quadrilateral (solid line) and triangular (dashed line) grid. Node vertex and fully viscous approach.

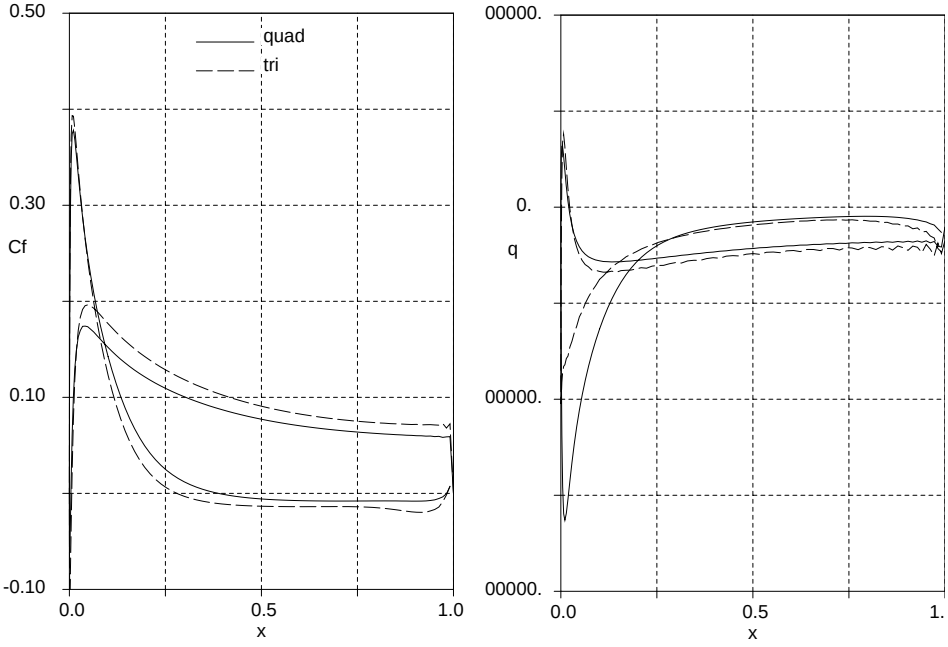


Fig. 7-16 Skin friction (left) and heat transfer distribution (right) with quadrilateral (solid line) and triangular (dashed line) grid. Node vertex and fully viscous approach.

7.2.2 Adiabatic wall

The rate of converge for the density residual and the integrated lift force can be seen in Figure 7-17 using an adiabatic wall boundary condition. The corresponding plots shown for the isothermal wall can be seen in Figure 7-18 to Figure 7-20. The results are very similar to the ones with the isothermal conditions. In general the results with the unstructured node vertex approach are close to the results with the structured cell centered approach. Any differences found with the isothermal conditions can be found here as well. The difference between the fully viscous and thin-layer approach that could be seen in the heat transfer only can here be seen in the temperature. No further results for this case using this boundary condition will therefore be shown.

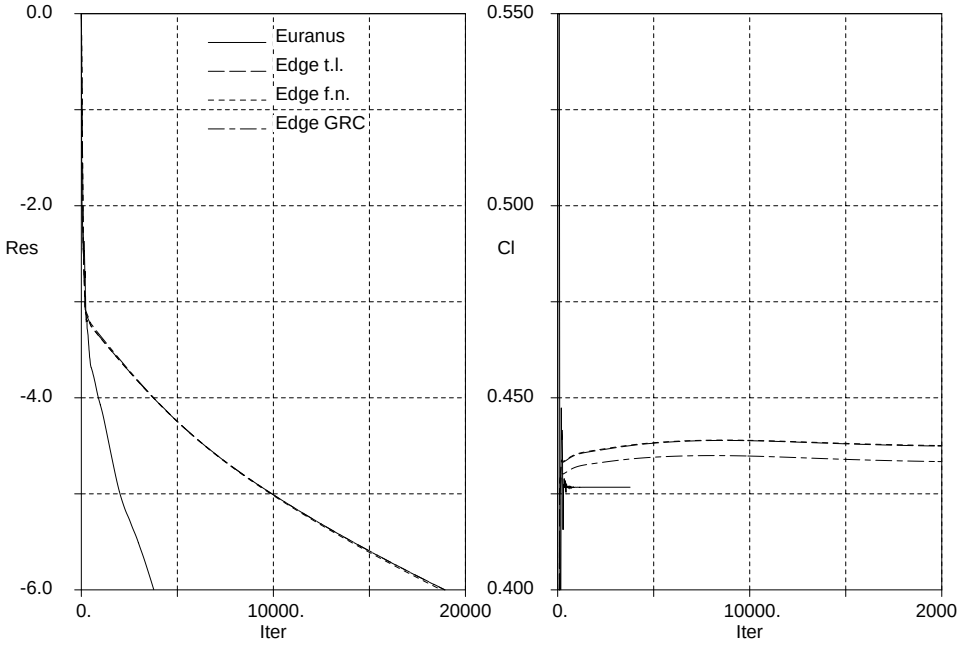


Fig. 7-17 Convergence of density residual and lift force. *EURANUS* fully viscous operator solid line, *EDGE* thin layer long dashes, *EDGE* fully viscous operator short dashes. Adiabatic wall conditions.

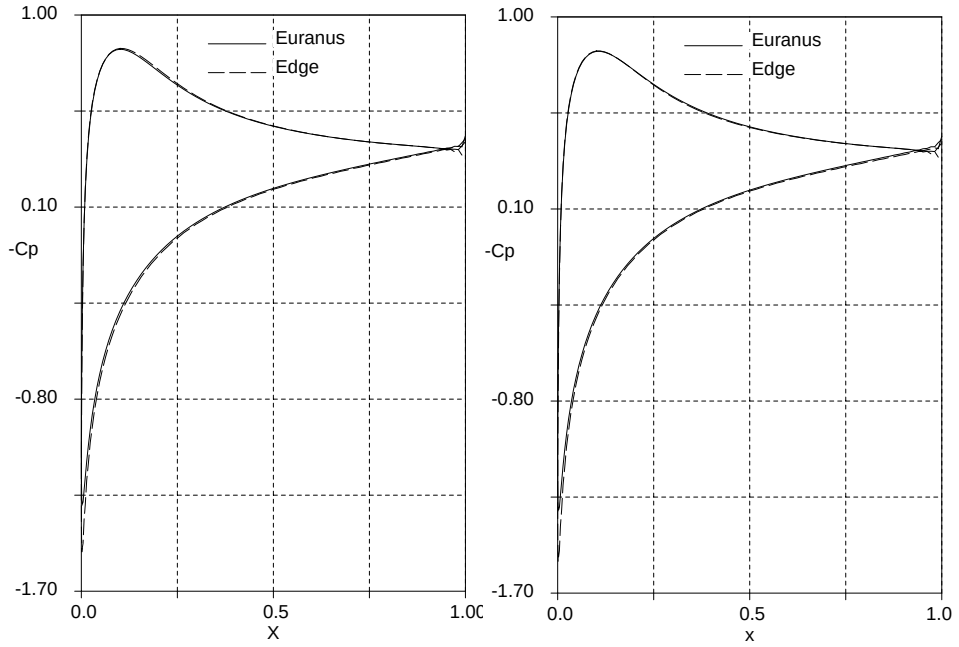


Fig. 7-18 Cp distribution. Left fully viscous operator, right thin-layer approach. *EURANUS* solid line, *EDGE* long dashes. Adiabatic wall conditions.

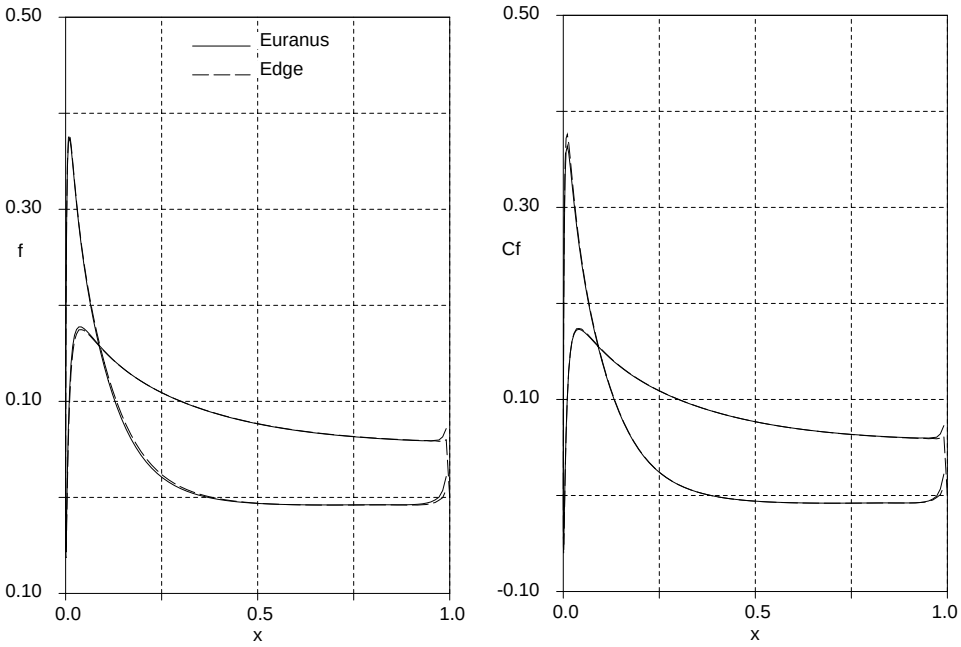


Fig. 7-19 Skin friction distribution. Left fully viscous operator, right thin-layer approach. *EURANUS* solid line, *EDGE* long dashes, *EDGE* with mod. gradients short dashes. Adiabatic wall conditions.

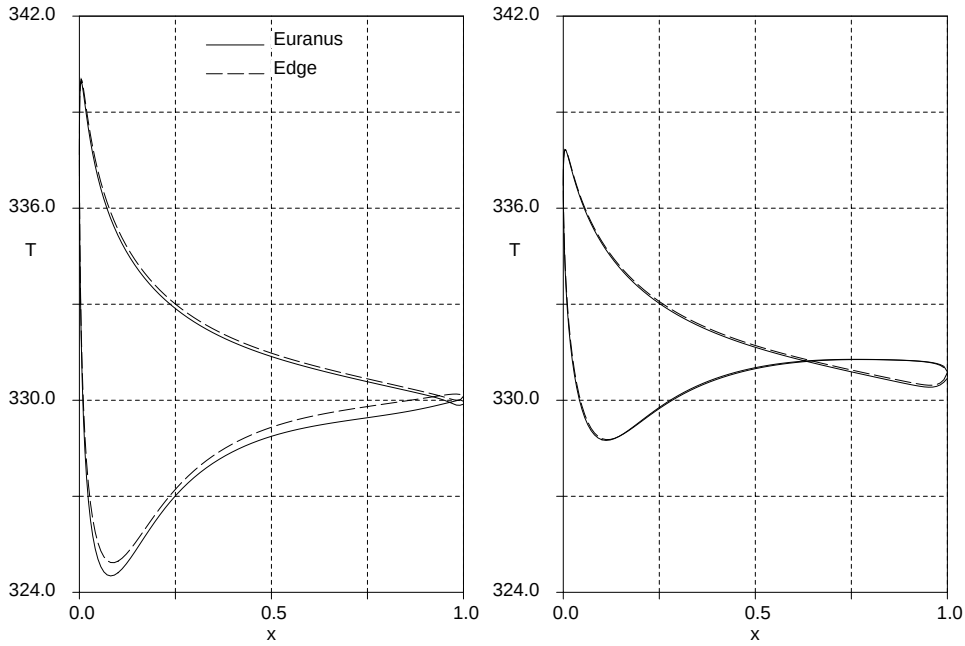


Fig. 7-20 Temperature distribution. Left fully viscous operator, right thin-layer approach. *EURANUS* solid line, *EDGE* long dashes, *EDGE* with mod. gradients short dashes. Adiabatic wall conditions.

7.3 Turbulent flow over the RAE2822 airfoil

A demanding test case is the transonic flow over a RAE2822 airfoil. This case has been subject for several earlier investigations in Euroval [21] and ECARP [22], the particular case investigated here is denoted *Case 9* and involves a shock interacting with a small boundary layer separation. The results focus on a comparison between numerical results with the node vertex/unstructured and cell centre/structured results and hence no experimental results are shown although available.

The flow conditions are

$$M = 0.734,\ \alpha = 2.79°,\ Re = 6.5\times10^{6}. \tag{7-4}$$

The trailing edge of the airfoil is sharp and hence a C-type of grid was used with a size of 273×81 nodes, 81 nodes from the wall to the outer boundary located about 10 chords away. A close up of the grid can be seen Figure 7-21. The distance to the second layer of nodes from the wall varies from 0.3×10^{-5} at the leading edge to 1×10^{-5} at the trailing edge. The computations are made fully turbulent although the location of the transition is known. The reason is that specification of transition was not implemented when the calculations were carried out.

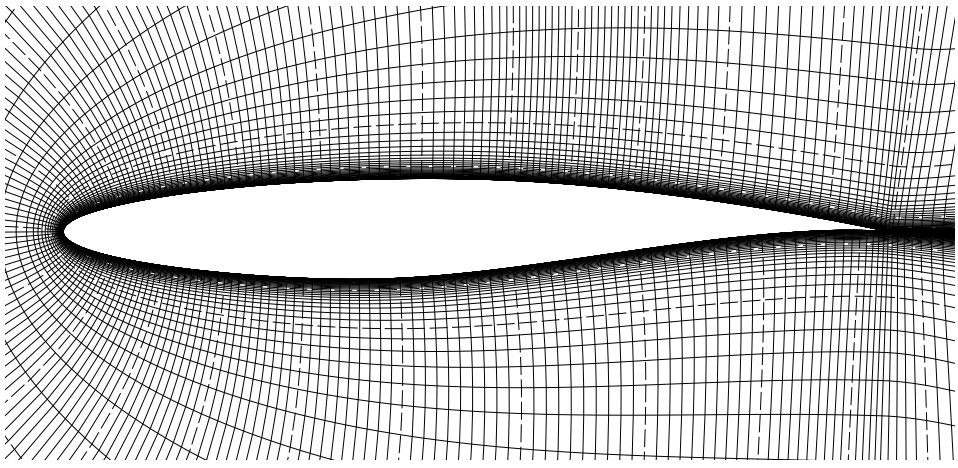


Fig. 7-21 RAE viscous grid of 273*81 grid points.

The same CFL number was used, CFL=1.5 accelerated with 4 levels of multigrid with V-cycles. The fully viscous terms were used. Similar values of the parameters for the artificial dissipation in the two solvers were chosen although the implementation is different. A second order upwind scheme for the turbulent equations was used for both methods.

The rate of convergence can be seen in Figure 7-22 for the structured cell centered and unstructured cell vertex approaches. The rate of convergence is similar.

The rate is also effected by the value of the numerical dissipation. The rate can be further improved by using full multigrid.

The pressure, temperature and skin friction distributions are compared in Figure 7-23 and Figure 7-24. In general there is a close resemblance between the results. There is some deviation for the temperature, and a small deviation for the skin friction. A small separation behind the shock is predicted in both cases.

Both approaches show some sensitivity in the results due to the setting of some of the numerical parameters. The parameters that influence the results are the level of numerical dissipation for the mean flow as well as the entropy fix for the upwind discretization of the turbulent equations (which also influences the level of artificial dissipation for the turbulence). The differences are rather small though.

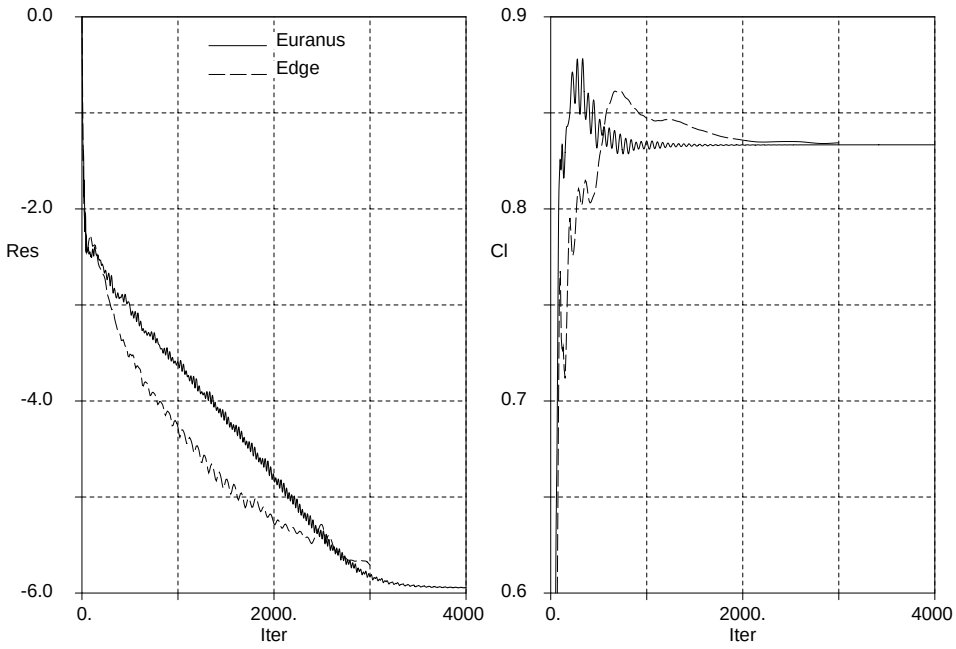


Fig. 7-22 Convergence of density residual (left) and lift force (right) for the turbulent RAE airfoil computation. *EURANUS* solid line, *EDGE* dashed line.

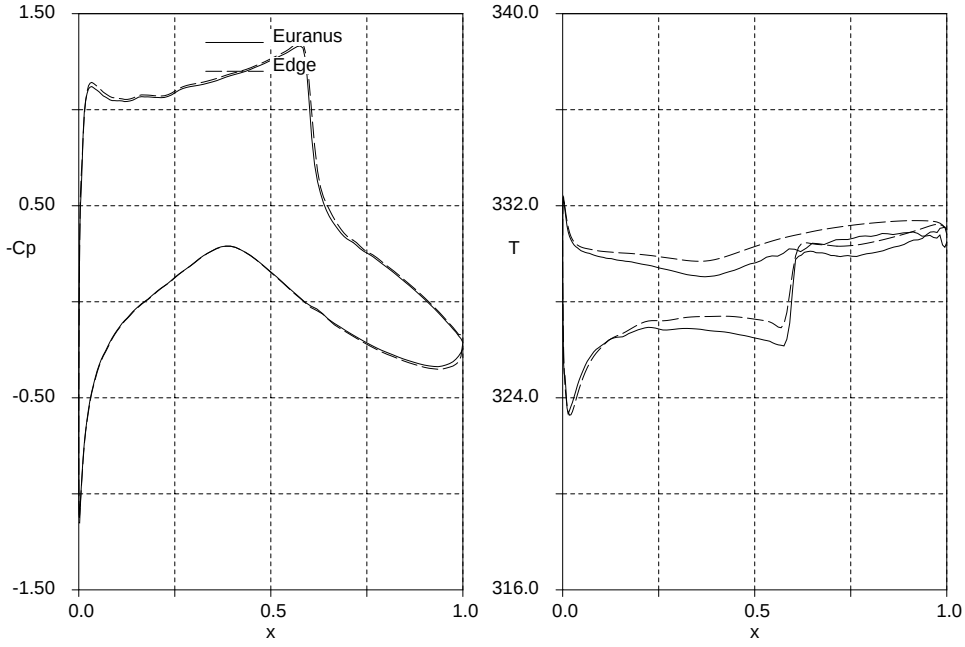


Fig. 7-23 Cp (left) and temperature distribution (right) for the turbulent RAE airfoil computation. *EURANUS* solid line, *EDGE* dashed line.

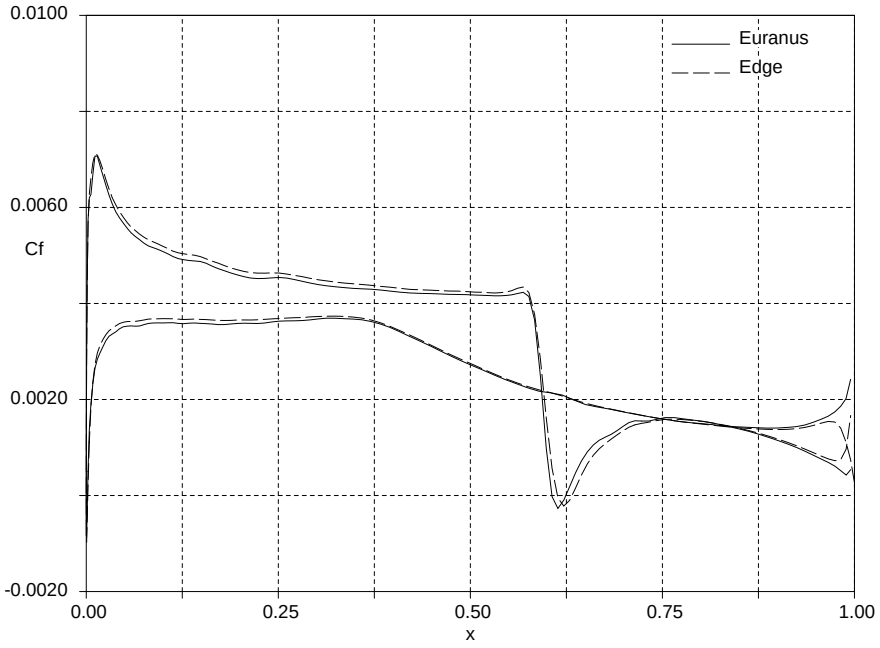


Fig. 7-24 Skin friction for the turbulent RAE airfoil computation. *EURANUS* solid line, *EDGE* dashed line.

7.4 Turbulent flow over the M6 wing

The last test case covers transonic flow over a finite 3D wing, namely the ONERA M6 wing. The flow conditions are

$$M = 0.84, \alpha = 3.06°, Re = 11.7\times10^{6}. \quad (7\text{-}5)$$

where the Reynolds number is based on the aerodynamic mean chord. Detailed experimental pressure distributions are available for this case [23] and there is also a variety of numerical results available. With a sufficient grid resolution this case is known to be rather insensitive to the turbulence model and also to produce results which agree fairly well with experiments.

A structured and an unstructured grid have been used in the evaluation. Only the unstructured grid has been used with *EDGE*. The unstructured grid has been adapted one time which can be seen on the surface grid in Figure 7-25. The adapted surface grid reveal that there are two shocks merged into one close to the tip. About 1.2 millions points is used for the structured O-type grid, about 0.9 million points for the unstructured grid. Both grids contain about 12.000 nodes on the surface, the distance from the surface to the first outer node is approximately the same and constant. The unstructured grid has prismatic elements in the boundary layer and tetrahedra in the rest of the domain.

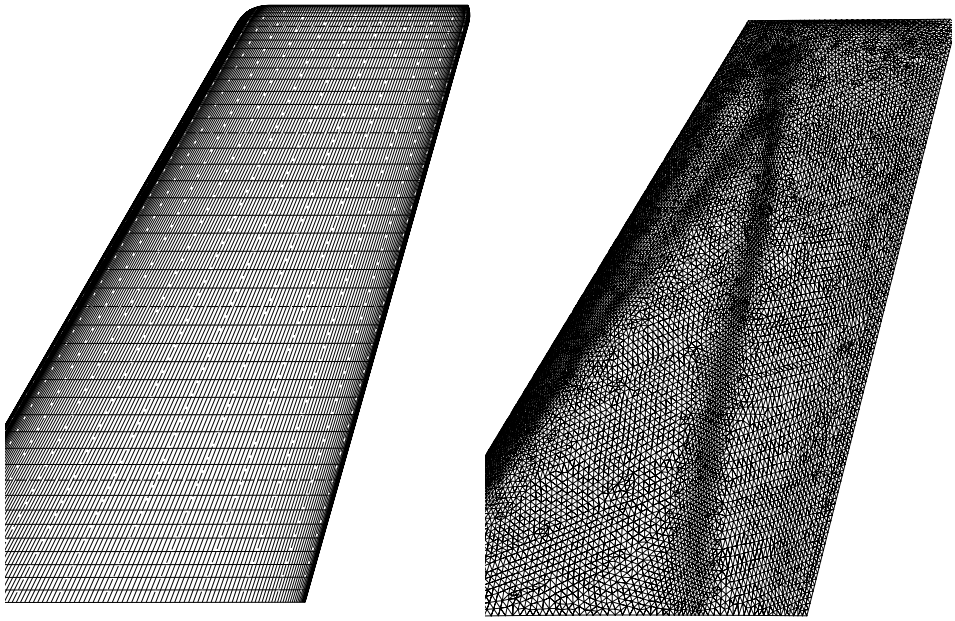


Fig. 7-25 Surface grid for the ONERA M6 wing. Structured grid to the left, unstructured grid to the right. Both grids contain about 12000 nodes on the surface. In total 1.2 millions points for the structured grid, 0.9 millions points for the unstructured grid.

The k-ω turbulence model was used. In addition *EDGE* used also an algebraic Reynolds stress model [5] to investigate the influence from the turbulence model. Five multigrid levels were used with the structured solver, four levels with the unstructured solver. In both cases full multigrid was used. The CFL numbers used were CFL=3.0 for the structured calculation and CFL=1.5 for the unstructured ones. The rate of convergence can be seen in Figure 7-26. The rate of convergence is comparable between the two approaches. The residuals drop faster with the unstructured approach, they drop about 4.5 orders of magnitude in only 500 multigrid cycles. The lift, however, converges very rapidly with the structured approach.

It should be mentioned that the coarser grids are generated differently. In the structured approach every second point is removed in all direction giving a coarsening ratio of 1:8. With the unstructured approach a coarsening ratio of 1:2 is applied in the prismatic region, i.e. cells are agglomerated only in the direction normal to the wall. In the tetrahedral region a coarsening ratio of 1:8 is applied. The resulting average coarsening ratio between the grids becomes about 1:3. It should also be noted that the convergence rate, although not plotted, is similar between the two different turbulence models.

The pressure distribution of the three numerical solutions compared to experimental results can be seen in Figure 7-27. The pressure is displayed at six different span wise locations, 20%, 44%, 65%, 80%, 90% 95% of the span.

The agreement with experimental data is fairly good and is what can be expected. The results with the unstructured solver are close to the results with the structured solver. *EDGE* produces slightly better results due to the increased grid resolution from the grid adaptation, especially in the tip region. Very small differences can be observed between the two turbulence models.

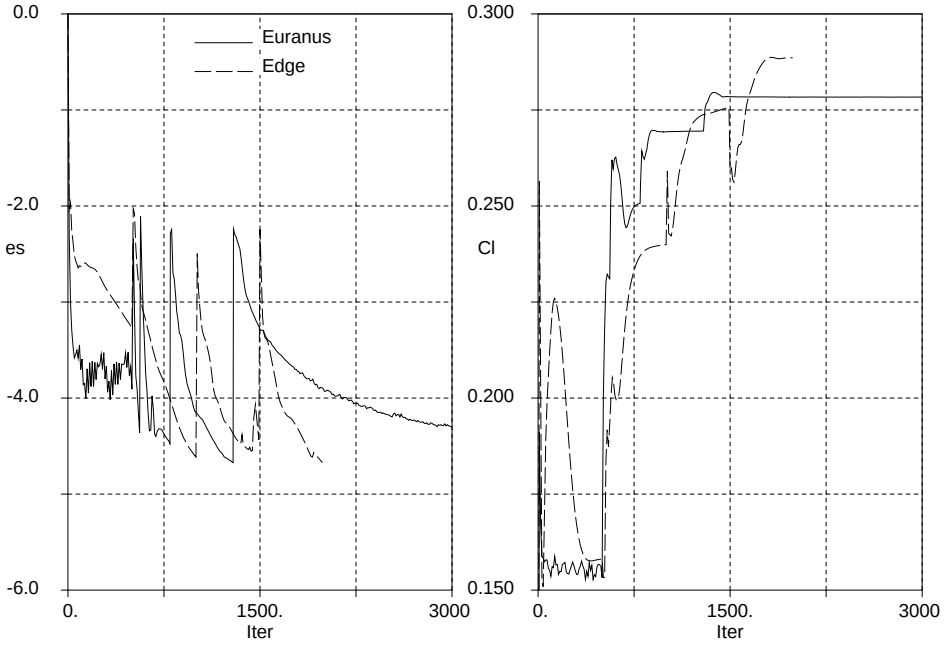


Fig. 7-26 Convergence of the residual (left) and lift (right). Structured results (solid line) obtained with 5 multigrid levels and full multigrid. 4 multigrid levels with the unstructured approach (dashed line).

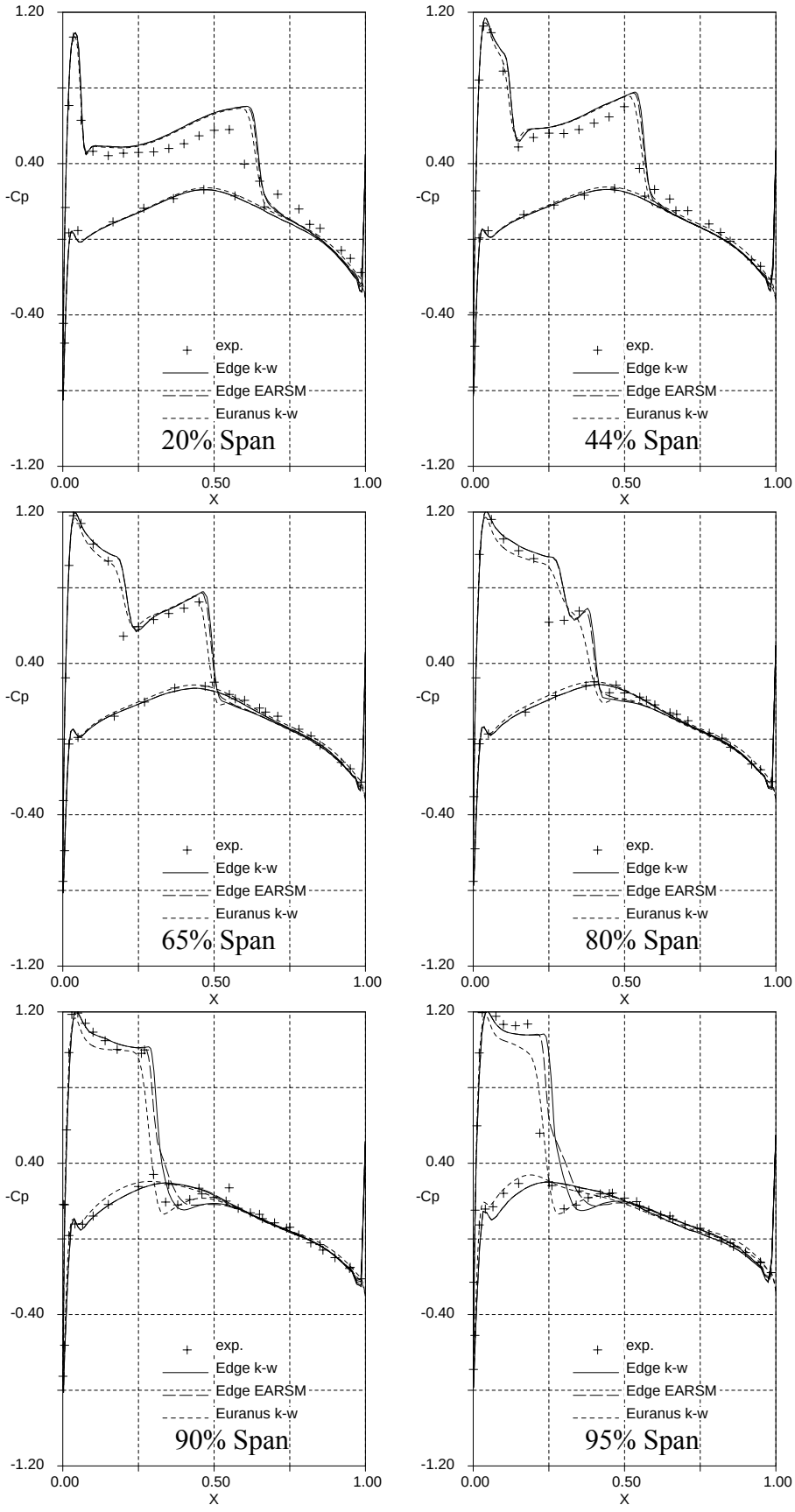


Fig. 7-27 Pressure distribution. Solid line - *EDGE* and k-ω, long dashes - *EDGE* and EARSM, small dashes - *EURANUS* and k-ω.

8 Summary and conclusions

The development and validation of a node-centered edge-based steady-state Reynolds averaged Navier-Stokes solver *EDGE* for unstructured grids is described in this report. The edge-based formulation in conjunction with a preprocessor that translates element-based information to edge-based information makes the system equally applicable to any type of element. It uses a node-centered finite volume scheme to solve the compressible equations in two or three space dimensions. It relies on explicit time integration with convergence acceleration by agglomeration multigrid and implicit residual smoothing. A second order accurate central scheme with artificial dissipation or an upwind TVD scheme with limiting of the characteristics may be used. Two compact discretizations of the viscous terms have been proposed and evaluated.

The validation has been carried out for external flows over airfoils and an ONERA M6 3D wing. An Euler case, a laminar low Reynolds number case and two turbulent cases has been used. More Euler validations have earlier been carried out [1]. The validations are focused on comparisons between the cell centered solver *EURANUS* on structured grids and the *EDGE* cell vertex solver on the same grid but in an unstructured format. In some cases the grid was triangulated to see the effect of different cell types.

There is a negligible difference between the two discretizations of the viscous terms. The computed results compare well with the results of the cell centered approach for structured grids. The rate of convergence has been found to be comparable for most cases. It is often possible though to take a somewhat higher CFL number with a solver for structured grids leading to a somewhat better rate of convergence.

There are some differences though that should be pointed out.

- The decay of the maximum total pressure loss for the subsonic Euler case is, for *EDGE*, about second order accurate as expected when the grid is refined. The order of accuracy depends to some extent on the spatial discretization and on the cell type. Interesting, though, is that the decay is only first order accurate with the structured cell centred approach.
- The rate of convergence for *EDGE* is much slower compared to *EURANUS* for the laminar low Reynolds number case without artificial dissipation. The solutions are similar though with the exception of a region close to the nose where there are small pressure oscillations for *EDGE*. A possible explanation for the difference could be the implementation of the boundary conditions. This has to be further examined.
- The differences for the turbulent flow are found at the trailing edge in particular. Remaining small differences can be explained by the level and different implementation of the numerical dissipation for the mean flow as well as for the turbulent equations.

References

[1] Sjögren, T. and Eliasson, P. (1998) “Description and Validation of EDGE, An Edge-based Euler Solver for Unstructured Grids”, FFA TN 1998-61.

[2] Berglind, T. “An Agglomeration Algorithm for Navier-Stokes Grids”, AIAA-2000-2254.

[3] Rizzi, A., Eliasson, P., Lindblad, I., Hirsch, C., Lacor, C. and Haeuser, J. (1993) “The Engineering of Multiblock \ Multigrid Software for Navier-Stokes Flows on Structured Meshes”, Computers and Fluids, Vol. 22, pp. 341-367.

[4] Wilcox, D.C (1993) “Turbulence Models for CFD”, DCW Industries Inc., La Canada, California.

[5] Wallin, S., and Johansson, A. (2000) “A complete explicit Algebraic Reynolds Stress Model for Incompressible and Compressible Turbulent Flows”, Journal of Fluid Mechanics, 403, pp 89-132.

[6] Berglind, T. (1994) “Finite Volume Solution of 2D Euler Equations on Unstructured Triangular Grids”, FFA TN 1994-17.

[7] Mavriplis, D. (1988) “Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes”, AIAA 88-3706-CP.

[8] Martinelli (1987) “Calculation of Viscous Flows with Multigrid Methods”, PhD Thesis, MAE Department, Princeton University

[9] Hirsch, C. (1990) “Numerical Computation of Internal and External Flows”, John Wiley & Sons, Vol. 2.

[10] Barth, T. and Jespersen, D. (1989) “The Design and Application of Upwind Schemes on Unstructured Meshes”, AIAA-89-0366.

[11] Peroomian, O., Chakravarthy, S. and Goldberg, U. (1997) “A Grid-Transparent Methodology for CFD”, AIAA-97-0724.

[12] Peroomian, O., Chakravarthy, S. (1996) “A Feasability Study of a Cell-Averaged-Based Multi-Dimensional ENO Scheme for use in Supersonic Shear Layers”, AIAA-96-0524.

[13] Weiss, J.M., Maruszewski, J.P. and Wayne, A.S. (1997) “Implicit Solution of the Navier-Stokes Equation on Unstructured Meshes”, AIAA 97-2103.

[14] Gnoffo, P.A. “An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows”, NASA TP-2953, 1990.

[15] Haaselbacher, A., McGuirk, J., and Page, G. (1999) “Finite Volume Discretization Aspects on Mixed Unstructured Grids”, AIAA Journal, vol. 37, No. 2.

[16] Barth, T.J. (1991) “Numerical Aspects of Computing Viscous High Reynolds Number Flows on Unstructured Meshes”, AIAA 91-0721.

[17] Radespiel, R. and Swanson, R.C. (1991) “Progress with Multigrid for Hypersonic Flow Problems, ICASE Report No. 91-89.

[18] Mavripilis (1989) “Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes”, AIAA-89-0120.

[19] Hellsten, A. (1998) “On the solid-wall boundary condition for ω in the k-ω - type turbulence modes”, Helsinki University of Technology, Report No B-50, Series B.

[20] Bristeau, O.B, Glowinski, R., Periaux, J. and Vivian, H. (1986) “Numerical Simulation of Compressible Navier-Stokes Flows”, Notes on Numerical Fluid Mechanics, Vol. 18, Vieweg.

[21] Haase, W., Bradsma, F., Elsholz, E., Leschziner, M. and Schwamborn, D. (1993) “*EUROVAL - An European Initiative on Validation of CFD Codes*”, Notes on Numerical Fluid Mechanics, Vol. 42, Vieweg Verlag.

[22] Haase, W., Chaput, E., Elsholz, E., Leschziner, M.A. and Mueller U.R. (1997) “*ECARP - European Computational Aerodynamics Research Project: Validation of CFD Codes and Assessment of Turbulence Models*”, Notes on Numerical Fluid Mechanics, Vol. 58, Vieweg Verlag.

[23] Schmitt, V. and Charpin, F. “Pressure Distributions on the ONERA-M6 Wing at Transonic Mach Numbers”, AGARD-AR-138.

Appendix A

A.1 3D inviscid Jacobians and Eigenvectors

The 3D Euler system is written as:

$$\frac{\partial}{\partial t}\bar{q} + \frac{\partial}{\partial x}\bar{F} + \frac{\partial}{\partial y}\bar{G} + \frac{\partial}{\partial z}\bar{H} = 0$$

where $\bar{q} = (\rho, \rho u, \rho v, \rho w, E)^T$ are the conservative variables. Introducing the Jacobian of the conservative variables A, B, C;

$$\frac{\partial}{\partial t}\bar{q} + A\frac{\partial}{\partial x}\bar{q} + B\frac{\partial}{\partial y}\bar{q} + C\frac{\partial}{\partial z}\bar{q} = 0$$

Alternatively the primitive variables may be used, denoted $\bar{v} = (\rho, u_1, u_2, u_3, p)^T$:

$$\frac{\partial}{\partial t}\bar{v} + \tilde{A}\frac{\partial}{\partial x}\bar{v} + \tilde{B}\frac{\partial}{\partial y}\bar{v} + \tilde{C}\frac{\partial}{\partial z}\bar{v} = 0$$

where $\tilde{A}, \tilde{B}, \tilde{C}$ are the Jacobians of the primitive variables, denoted non-conservative Jacobians. With the transformation matrix

$$M = \frac{\partial}{\partial \bar{v}}\bar{q}$$

between primitive and conservative variables the following relations between the Jacobians hold:

$$\tilde{A} = M^{-1}AM, \tilde{B} = M^{-1}BM, \tilde{C} = M^{-1}CM$$

Introducing the notation

$$A_n = n_x A + n_y B + n_z C$$

the conservative Jacobians can be made diagonal as follows:

$$A_n = R\Lambda R^{-1}$$

and the non-conservative Jacobians

$$\tilde{A}_n = n_x\tilde{A} + n_y\tilde{B} + n_z\tilde{C}$$

can be made diagonal as

$$\tilde{A}_n = L\Lambda L^{-1}$$

The diagonal matrix Λ contains the eigenvalues to the Jacobians $A_n, \tilde{A}_n$, the matrices R, L contain the right eigenvectors as columns to the conservative and non-conservative Jacobians, respectively. The eigenvectors given below are scaled in such a way that the characteristic variables have the dimension of a pressure. Note that the scaling of the eigenvectors is not unique, nor is the choice of the eigenvectors.

The matrices are the following:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u_1 & \rho & 0 & 0 & 0 \\ u_2 & 0 & \rho & 0 & 0 \\ u_3 & 0 & 0 & \rho & 0 \\ \frac{\bar{u}^2}{2} & \rho u_1 & \rho u_2 & \rho u_3 & \frac{1}{\gamma-1} \end{bmatrix}, M^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -\frac{u_1}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 \\ -\frac{u_2}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 \\ -\frac{u_3}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 \\ (\gamma-1)\frac{\bar{u}^2}{2} & -(\gamma-1)u_1 & -(\gamma-1)u_2 & -(\gamma-1)u_3 & \gamma-1 \end{bmatrix}$$

$$L = \begin{bmatrix} \frac{n_x}{c^2} & \frac{n_y}{c^2} & \frac{n_z}{c^2} & \frac{1}{c^2} & \frac{1}{c^2} \\ 0 & -\frac{n_z}{\rho c} & \frac{n_y}{\rho c} & \frac{n_x}{\rho c} & -\frac{n_x}{\rho c} \\ \frac{n_z}{\rho c} & 0 & -\frac{n_x}{\rho c} & \frac{n_y}{\rho c} & -\frac{n_y}{\rho c} \\ -\frac{n_y}{\rho c} & \frac{n_x}{\rho c} & 0 & \frac{n_z}{\rho c} & -\frac{n_z}{\rho c} \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}, L^{-1} = \begin{bmatrix} n_x c^2 & 0 & n_z \rho c & -n_y \rho c & -n_x \\ n_y c^2 & -n_z \rho c & 0 & n_x \rho c & -n_y \\ n_z c^2 & n_y \rho c & -n_x \rho c & 0 & -n_z \\ 0 & \frac{n_x \rho c}{2} & \frac{n_y \rho c}{2} & \frac{n_z \rho c}{2} & \frac{1}{2} \\ 0 & -\frac{n_x \rho c}{2} & -\frac{n_y \rho c}{2} & -\frac{n_z \rho c}{2} & \frac{1}{2} \end{bmatrix}$$

$$\Lambda = \begin{bmatrix} \bar{u} \cdot \bar{n} & & & & \\ & \bar{u} \cdot \bar{n} & & & \\ & & \bar{u} \cdot \bar{n} & & \\ & & & \bar{u} \cdot \bar{n} + c & \\ & & & & \bar{u} \cdot \bar{n} - c \end{bmatrix}$$

$$R = ML =$$

$$\begin{bmatrix} \frac{n_x}{c^2} & \frac{n_y}{c^2} & \frac{n_z}{c^2} & \frac{1}{c^2} & \frac{1}{c^2} \\ \frac{u_1 n_x}{c^2} & \frac{u_1 n_y - c n_z}{c^2} & \frac{u_1 n_z + c n_y}{c^2} & \frac{u_1 + c n_x}{c^2} & \frac{u_1 - c n_x}{c^2} \\ \frac{u_2 n_x + c n_z}{c^2} & \frac{u_2 n_y}{c^2} & \frac{u_2 n_z - c n_x}{c^2} & \frac{u_2 + c n_y}{c^2} & \frac{u_2 - c n_y}{c^2} \\ \frac{u_3 n_z - c n_y}{c^2} & \frac{u_3 n_y + c n_x}{c^2} & \frac{u_3 n_z}{c^2} & \frac{u_3 + c n_z}{c^2} & \frac{u_3 - c n_z}{c^2} \\ n_x \frac{\bar{u}^2}{2c^2} + \frac{u_2 n_z - u_3 n_y}{c} & n_y \frac{\bar{u}^2}{2c^2} + \frac{u_3 n_x - u n_z}{c} & n_z \frac{\bar{u}^2}{2c^2} + \frac{u_1 n_y - v n_x}{c} & \frac{H + c(\bar{u} \cdot \bar{n})}{c^2} & \frac{H - c(\bar{u} \cdot \bar{n})}{c^2} \end{bmatrix}$$

$$R^{-1} =$$

$$\begin{bmatrix}
n_x c^2\left(1-\frac{\gamma-1}{2}M^2\right) & (\gamma-1)u_1 n_x & (\gamma-1)u_2 n_x + n_z c & (\gamma-1)u_3 n_x - n_y c & -(\gamma-1)n_x \\
-c(u_2 n_z - u_3 n_y) & & & & \\
n_y c^2\left(1-\frac{\gamma-1}{2}M^2\right) & (\gamma-1)u_1 n_y - n_z c & (\gamma-1)u_2 n_y & (\gamma-1)u_3 n_y + n_x c & -(\gamma-1)n_y \\
-c(u_3 n_x - u_1 n_z) & & & & \\
n_z c^2\left(1-\frac{\gamma-1}{2}M^2\right) & (\gamma-1)u_1 n_z + n_y c & (\gamma-1)u_2 n_z - n_x c & (\gamma-1)u_3 n_z & -(\gamma-1)n_z \\
-c(u_1 n_y - u_2 n_x) & & & & \\
\frac{c^2}{2}\left(\frac{\gamma-1}{2}M^2 - \frac{\bar{u}\cdot\bar{n}}{c}\right) & \frac{c n_x - (\gamma-1)u_1}{2} & \frac{c n_y - (\gamma-1)u_2}{2} & \frac{c n_z - (\gamma-1)u_3}{2} & \frac{\gamma-1}{2} \\
\frac{c^2}{2}\left(\frac{\gamma-1}{2}M^2 + \frac{\bar{u}\cdot\bar{n}}{c}\right) & \frac{-c n_x - (\gamma-1)u_1}{2} & \frac{-c n_y - (\gamma-1)u_2}{2} & \frac{-c n_z - (\gamma-1)u_3}{2} & \frac{\gamma-1}{2}
\end{bmatrix}$$

where H is the total enthalpy

$$H = \frac{E+p}{\rho} = \frac{c^2}{\gamma-1} + \frac{1}{2}\bar{u}^2$$

and c the speed of sound and $M = |\bar{u}|/c$ the Mach number.

Issuing organization
FOI-Swedish Defence Research Agency
Division of Aeronautics, FFA
SE-172 90 STOCKHOLM

Report number, ISRN
FOI-R--0298--SE

Report type
Scientific report

Month year
December 2001

Project number
A84 0297

Customer code
3. Aeronautical Research

Research area code
7. Vehicles

Sub area code
73. Aeronautical Research

Author(s)
Peter Eliasson

Project manager
Peter Eliasson

Approved by
Torsten Berglind
Head, Computational Aerodynamics Department

Scientifically and technically responsible
Jan Nordström
Research leader

Report title
EDGE, a Navier-Stokes Solver for Unstructured Grids

Abstract
This report describes the compressible Navier-Stokes solver *EDGE* for unstructured grids. The solver is based on an edge-based formulation for arbitrary elements and uses a node-centered finite-volume technique to solve the governing equations. Two spatial discretizations of the convection terms are described, compact discretizations of the thin-layer and fully viscous terms have been proposed and evaluated. The governing equations are integrated explicitly towards steady state with Runge-Kutta time integration. The convergence is accelerated with agglomeration multigrid and implicit residual smoothing.

A validation is carried out in two and three dimensions for external flows. The validations focus on comparisons between *EDGE* and the cell centered solver *EURANUS* on structured grids. Also the effect of different types of elements are investigated.

The results with the unstructured and structured approach compare well for all cases. The rate of convergence is comparable although higher CFL numbers can be used with the structured solver. The robustness of the unstructured solver is at least as good as with the structured solver.

Two main differences are found. The first is that the decay of the maximum total pressure loss for subsonic Euler calculations is approximately second order accurate as the grid is refined for the node-centered scheme but only first order using the structured cell centered approach. The second difference concerns the flow over an airfoil at a low Reynolds number and no artificial dissipation. Here the rate of convergence is much slower with the unstructured approach. Small oscillations in the pressure can also be observed in the nose region.

Keywords
CFD, unstructured grids, finite-volume, multigrid, edge formulation

Further bibliographic information

ISSN
ISSN 1650-1942

Pages
67

Language
English

Distribution according to missiv

Price
Price acc. to price list

Security classification
Unclassified

Utgivare	Rapportnummer, ISRN	Klassificering
Totalförsvarets Forskningsinstitut - FOI Avdelningen för Flygteknik, FFA SE-172 90 STOCKHOLM	FOI-R--0298--SE	Vetenskaplig rapport
	Månad år December 2001	Projektnummer A84 0297
	Verksamhetsgren 3. Flygteknisk forskning	
	Forskningsområde 7. Bemannade och obemannade farkoster	
	Delområde 73. Flygteknisk forskning	

Författare	Projektledare
Peter Eliasson	Peter Eliasson
	Godkänd av Torsten Berglind Chef, Institutionen för beräkningsaerodynamik
	Tekniskt och/eller vetenskapligt ansvarig Jan Nordström Forskningsledare

Rapporttitel

EDGE, en strömningslösare för ostrukturerade nät

Sammanfattning

Den här rapporten beskriver strömningslösaren *EDGE* för ostrukturerade nät. Lösaren är baserad på en kantbaserad framställning för godtyckliga element och använder en nodcentrerad teknik att lösa de kompressibel Navier-Stokes ekvationer. Två rumsdiskretiseringar av konvektionen beskrivs, två kompakta rumsdskretiseringa av de viskösa termerna har utvärderats. De underliggande ekvationerna integreras explicit i tiden med en Runge-Kutta metod till konvergens. Konvergensen accelereras med multigrid och residualutjämning.

Valideringen har utförts i två och tre dimensioner för extern strömning. Valideringarna fokuserar på jämförelser med den cell centrerade strömningslösaren *EURANUS* för strukturerade nät. Effekten av olika elementtyper har undersökts.

Resultaten med *EDGE* stämmer bra överens med resultaten med *EURANUS*. Konvergenshastigeheten är fullt jämförbar, dock kan man med *EURANUS* använda större CFL-tal. Robustheten är väl så bra *EDGE* med som med *EURANUS*.

Två skillnader kan dock noteras. Den första är att den maximala totaltrycksförlusten vid subsoniska Eulerberäkningar på en vingprofil där nätet successivt förfinas avtar ungefär med andra ordningens noggrannhet med *EDGE* medan *EURANUS* avtar med första ordningens noggrannhet. Den andra skillnaden är att vid viskös strömning för ett lågt Reynolds tal utan artificiell dissipation så är konvergenshatsighete lägre med *EDGE* än med *EURANUS*. Små svängningar i trycket kan också noters vid framkanten.

Nyckelord

CFD, ostrukturerade nät, finit volym, multigrid, kantbaserad formulering

Övriga bibliografiska uppgifter

ISSN	Antal sidor	Språk
ISSN 1650-1942	67	Engelska
Distribution enligt missiv Distribution	Pris Enligt prislista	
	Sekretess Öppen	